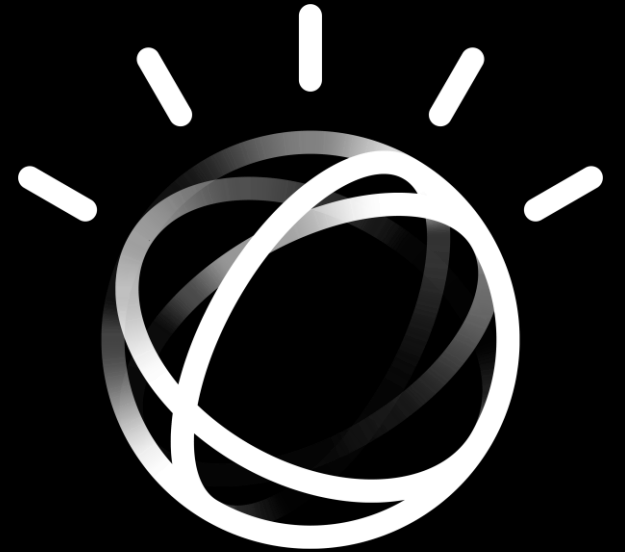


IBM Watson AIOps

Log Anomaly Detection

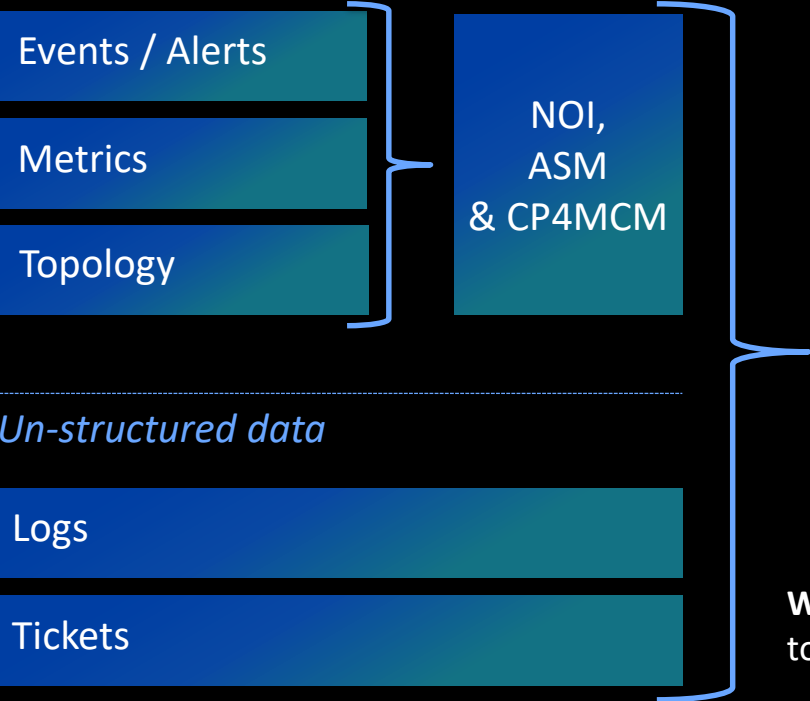
Leverage Watson AIOps to better predict IT outages from logs.



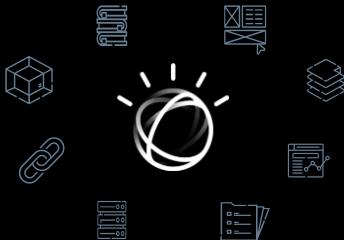
Watson AIOps

Data channels

Structured data

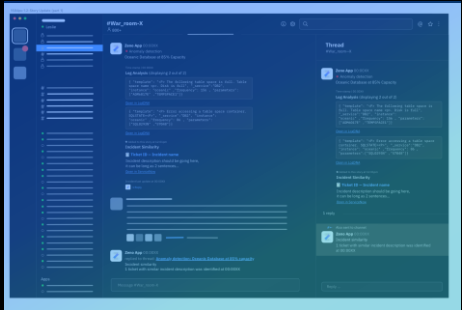


Watson AIOps



Insights surfaced via

ChatOps



Insights
& Advise

APIs for process
or dashboard integration
(future)



Watson AIOps integrates with **Netcool Operations Insights** to ingest events and combine it with unstructured data

Watson AIOps retrieves topology information from **ASM** to map its understanding of location and blast radius.

Watson AIOps retrieves additional data via integrations with **CP4MCM** to deliver additional insights & Next-Best-Actions

Logs



Find a View

⚡ EVERYTHING

VIEWS

499error-ts-ui-dashbo... 🔔

500 Internal Server Er... 🔔

500error-ts-ui-dashbo... 🔔

503 Service Unavailab... 🔔

demo-view

train-ticket-microserv...

ts-assurance-mongo-...

ts-services

ts-services+UI

ui-view



^ 66d9d8fb-ff78-48de-b1ac-...

IBM-7DAY

⚡ Everything ▾

🏷 All Tags ▾

📄 All Sources ▾

🏠 All Apps ▾

👤 All Levels ▾

```
Oct 18 21:07:11 kube-bqgas6od0kdun6p7q1lg-simulationd-wp16cpu-00000374 containerd.log time="2020-10-19T04:07:11.177897187Z" level=info msg="ExecSync for
\"cd400afaecf1b873cef5431e9e9409bd7338beef2c1fcbf26a8c40af1cb22d2\" with command [/usr/local/bin/galley probe --probe-path=/tmp/healthliveness --interval=10s] and timeout 1 (s)"
Oct 18 21:07:11 kube-bqgas6od0kdun6p7q1lg-simulationd-wp16cpu-00000374 containerd.log time="2020-10-19T04:07:11.206107012Z" level=info msg="ExecSync for
\"cd400afaecf1b873cef5431e9e9409bd7338beef2c1fcbf26a8c40af1cb22d2\" with command [/usr/local/bin/galley probe --probe-path=/tmp/healthready --interval=10s] and timeout 1 (s)"
Oct 18 21:07:11 weave-scope-agent-c5hzf scope-agent <probe> ERROR: 2020/10/19 04:07:11.793395 docker registry: cannot connect to Docker endpoint
Oct 18 21:07:11 kube-bqgas6od0kdun6p7q1lg-simulationd-wp16cpu-00000471 containerd.log time="2020-10-19T04:07:10.841377323Z" level=info msg="ExecSync for
\"493498c76d4d08e74bba4b476713ad60074cdb6d7a299d95c56a58bccdbf60ac\" with command [/usr/local/bin/sidecar-injector probe --probe-path=/tmp/health --interval=4s] and timeout 1 (s)"
Oct 18 21:07:11 kube-bqgas6od0kdun6p7q1lg-simulationd-wp16cpu-00000471 containerd.log time="2020-10-19T04:07:10.923327549Z" level=info msg="Finish piping \"stdout\" of container exec
\"5c3788bcf4d0a9a05ef3c2ccc2b80441fd00439577d29f3fb419b02251391b8e\""
Oct 18 21:07:11 kube-bqgas6od0kdun6p7q1lg-simulationd-wp16cpu-00000471 containerd.log time="2020-10-19T04:07:10.923476276Z" level=info msg="Finish piping \"stderr\" of container exec
\"5c3788bcf4d0a9a05ef3c2ccc2b80441fd00439577d29f3fb419b02251391b8e\""
Oct 18 21:07:11 kube-bqgas6od0kdun6p7q1lg-simulationd-wp16cpu-00000471 containerd.log time="2020-10-19T04:07:10.923779640Z" level=info msg="Exec process
\"5c3788bcf4d0a9a05ef3c2ccc2b80441fd00439577d29f3fb419b02251391b8e" exits with exit code 0 and error <nil>"
Oct 18 21:07:11 kube-bqgas6od0kdun6p7q1lg-simulationd-wp16cpu-00000471 containerd.log time="2020-10-19T04:07:10.925528943Z" level=info msg="ExecSync for
\"493498c76d4d08e74bba4b476713ad60074cdb6d7a299d95c56a58bccdbf60ac\" returns with exit code 0"
Oct 18 21:07:12 ts-assurance-mongo-6f7dbc4666-btvj2 ts-assurance-mongo I NETWORK [listener] connection accepted from 127.0.0.1:42802 #1667990 (2 connections now open)
Oct 18 21:07:12 ts-assurance-mongo-6f7dbc4666-btvj2 ts-assurance-mongo I NETWORK [conn1667990] received client metadata from 127.0.0.1:42802 conn1667990: { driver: { name: "PyMongo",
version: "3.8.0" }, os: { type: "Linux", name: "Linux", architecture: "x86_64", version: "4.15.0-96-generic" }, platform: "CPython 3.7.3.final.0" }
Oct 18 21:07:12 ts-assurance-mongo-6f7dbc4666-btvj2 ts-assurance-mongo I NETWORK [listener] connection accepted from 127.0.0.1:42804 #1667991 (3 connections now open)
Oct 18 21:07:12 ts-assurance-mongo-6f7dbc4666-btvj2 ts-assurance-mongo I NETWORK [conn1667991] received client metadata from 127.0.0.1:42804 conn1667991: { driver: { name: "PyMongo",
version: "3.8.0" }, os: { type: "Linux", name: "Linux", architecture: "x86_64", version: "4.15.0-96-generic" }, platform: "CPython 3.7.3.final.0" }
Oct 18 21:07:12 ts-assurance-mongo-6f7dbc4666-btvj2 ts-assurance-mongo I NETWORK [conn1667991] end connection 127.0.0.1:42804 (2 connections now open)
Oct 18 21:07:12 ts-assurance-mongo-6f7dbc4666-btvj2 ts-assurance-mongo I NETWORK [conn1667990] end connection 127.0.0.1:42802 (1 connection now open)
Oct 18 21:07:12 kube-bqgas6od0kdun6p7q1lg-simulationd-wp16cpu-00000374 containerd.log time="2020-10-19T04:07:11.790746500Z" level=info msg="Finish piping \"stdout\" of container exec
\"840af0a4bb97ee4fb62b9532d3d364516228ea5a4068ff893dae4c7ff0b009f4\""
Oct 18 21:07:12 kube-bqgas6od0kdun6p7q1lg-simulationd-wp16cpu-00000374 containerd.log time="2020-10-19T04:07:11.790909233Z" level=info msg="Finish piping \"stderr\" of container exec
\"840af0a4bb97ee4fb62b9532d3d364516228ea5a4068ff893dae4c7ff0b009f4\""
Oct 18 21:07:12 kube-bqgas6od0kdun6p7q1lg-simulationd-wp16cpu-00000374 containerd.log time="2020-10-19T04:07:11.790955580Z" level=info msg="Exec process
\"840af0a4bb97ee4fb62b9532d3d364516228ea5a4068ff893dae4c7ff0b009f4" exits with exit code 0 and error <nil>"
Oct 18 21:07:12 kube-bqgas6od0kdun6p7q1lg-simulationd-wp16cpu-00000374 containerd.log time="2020-10-19T04:07:11.828988179Z" level=info msg="ExecSync for
\"cd400afaecf1b873cef5431e9e9409bd7338beef2c1fcbf26a8c40af1cb22d2\" returns with exit code 0"
Oct 18 21:07:12 kube-bqgas6od0kdun6p7q1lg-simulationd-wp16cpu-00000374 containerd.log time="2020-10-19T04:07:11.858755954Z" level=info msg="Finish piping \"stdout\" of container exec
\"b7ecebb61e723ba1308ed33002cc7ab40c385b4ed36c02f21917863144014f7f\""
Oct 18 21:07:12 kube-bqgas6od0kdun6p7q1lg-simulationd-wp16cpu-00000374 containerd.log time="2020-10-19T04:07:11.858872118Z" level=info msg="Finish piping \"stderr\" of container exec
\"b7ecebb61e723ba1308ed33002cc7ab40c385b4ed36c02f21917863144014f7f\""
Oct 18 21:07:12 kube-bqgas6od0kdun6p7q1lg-simulationd-wp16cpu-00000374 containerd.log time="2020-10-19T04:07:11.858967658Z" level=info msg="Exec process
\"b7ecebb61e723ba1308ed33002cc7ab40c385b4ed36c02f21917863144014f7f" exits with exit code 0 and error <nil>"
Oct 18 21:07:12 kube-bqgas6od0kdun6p7q1lg-simulationd-wp16cpu-00000374 containerd.log time="2020-10-19T04:07:11.860817554Z" level=info msg="ExecSync for
\"cd400afaecf1b873cef5431e9e9409bd7338beef2c1fcbf26a8c40af1cb22d2\" returns with exit code 0"
Oct 18 21:07:12 ibm-master-proxy-static-10.95.163.216 ibm-master-proxy-static Oct 19 04:07:12 kube-bqgas6od0kdun6p7q1lg-simulationd-wp32cpu-0000051e local0.info haproxy[1]:
172.20.0.1:16433 [19/Oct/2020:04:07:12.805] masterapiserverfrontend masterapiserverbackend/c2-3.us-south.containers.cloud.ibm.com 1/-/1 0 CC 24/20/19/10/0 0/0
```



Search...



Jump to timeframe



● LIVE

Demo

Why Watson AIOps?

Monitoring Metrics Is Not Enough.
Keyword Query Is Not Enough.
Requiring Data Scientists Is Too Much.



```
cugraph_algo.ipynb
Python 3

Deep Learning Toolkit for Splunk - Graph Algorithms with cuGraph

This notebook contains examples for graph algorithms available in cuGraph

Note: By default every time you save this notebook the cells are exported into a python module which is then invoked by Splunk MLTK commands like | fit ... | apply ... | summary . Please read the Model Development Guide in the Deep Learning Toolkit app for more information.

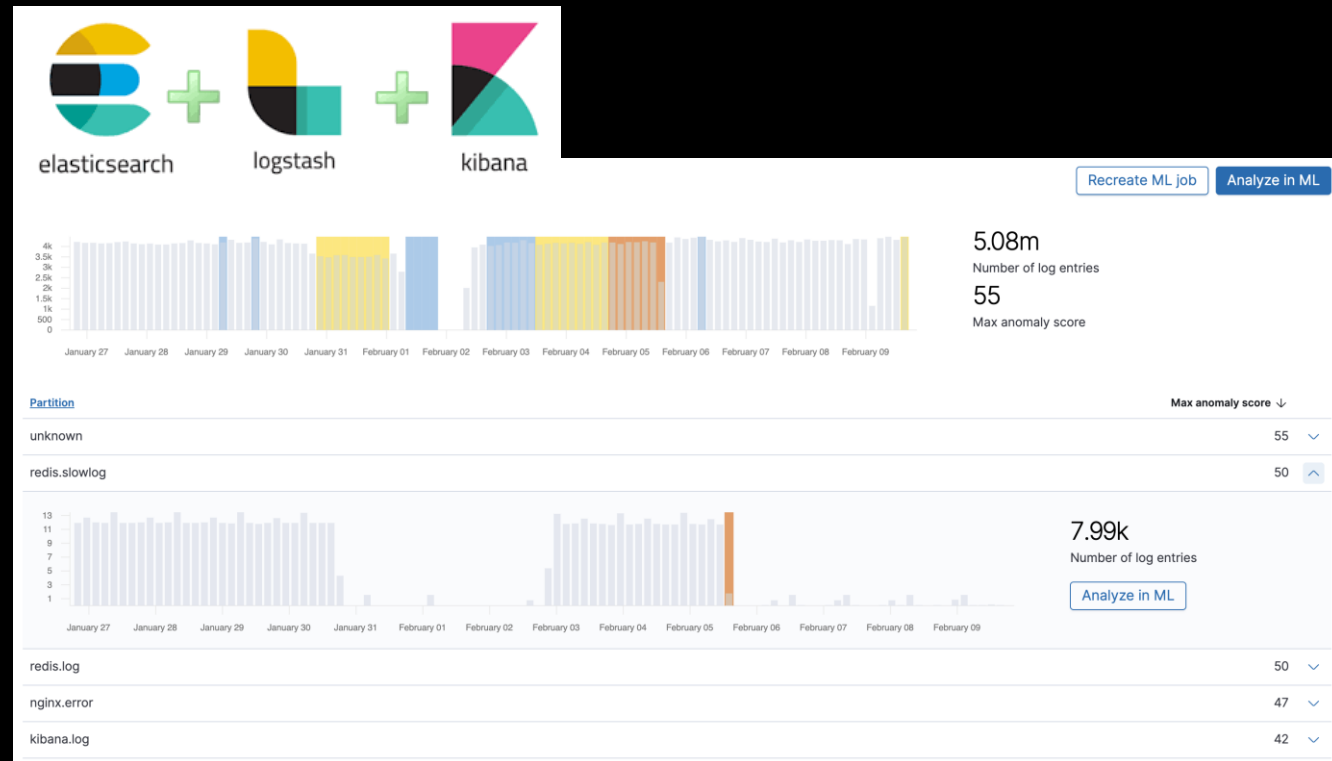
Stage 0 - import libraries

At stage 0 we define all imports necessary to run our subsequent code depending on various libraries.

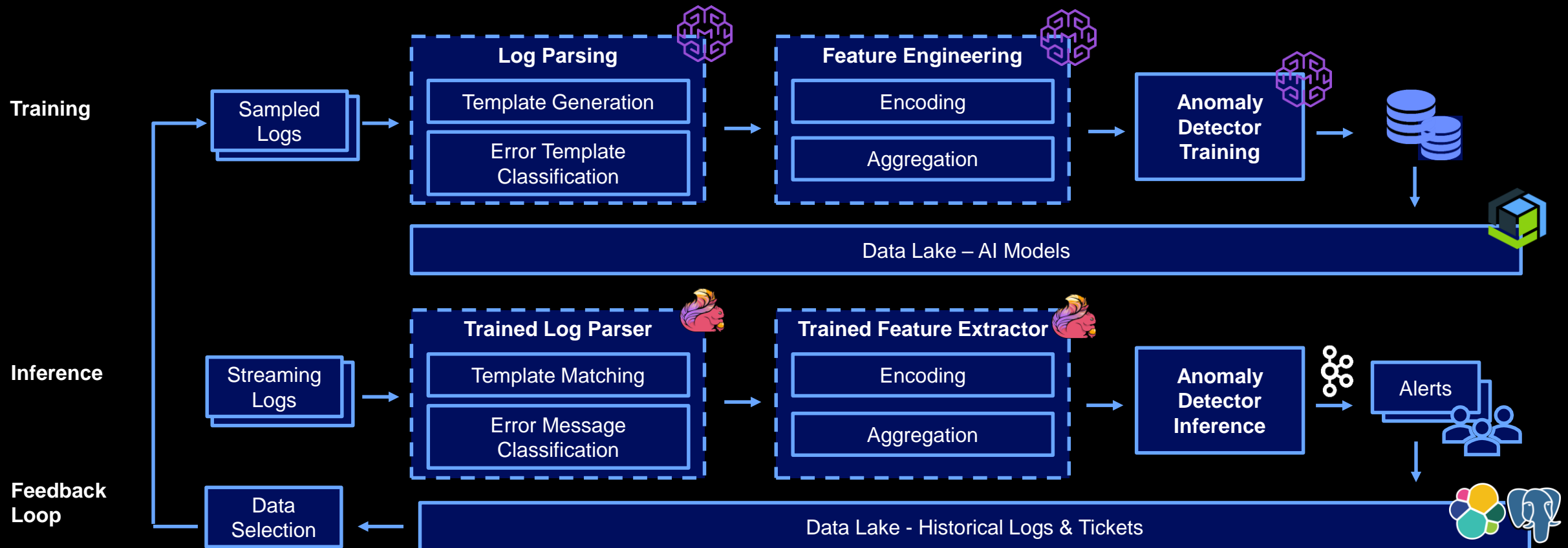
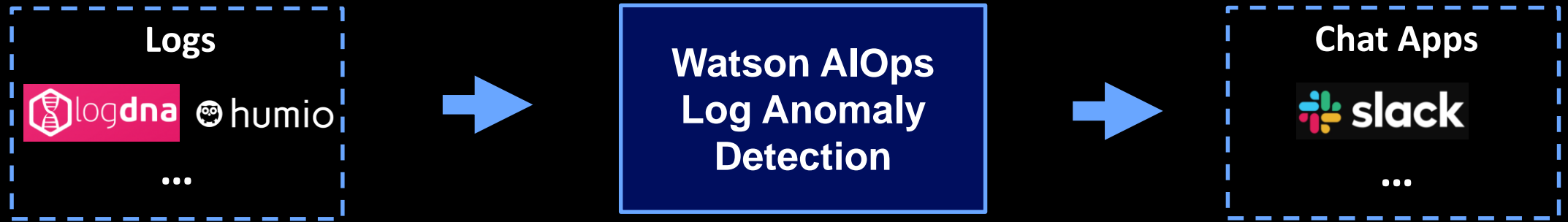
[1]: # this definition exposes all python module imports that should be available in all subsequent commands
import json
import numpy as np
import cudf as cd
import cugraph as cg
# ...
# global constants
MODEL_DIRECTORY = "/srv/app/model/data/"

[2]: # THIS CELL IS NOT EXPORTED - free notebook cell for testing or development purposes
print("numpy version: " + np.__version__)
print("cudf version: " + cd.__version__)
print("cugraph version: " + cg.__version__)

numpy version: 1.16.4
cudf version: 0.10.0
cugraph version: 0.10.0+0.ged867e5.dirty
```



Log Anomaly Detection



Template Generation

```
/* A logging code snippet extracted from:  
hadoop/hdfs/server/datanode/BlockReceiver.java */  
  
LOG.info("Received block " + block + " of size "  
+ block.getNumBytes() + " from " + inAddr);
```



Log Message

```
2015-10-18 18:05:29,570 INFO dfs.DataNode$PacketResponder: Received  
block blk_-562725280853087685 of size 67108864 from /10.251.91.84
```



Structured Log

TIMESTAMP	2015-10-18 18:05:29,570
LEVEL	INFO
COMPONENT	dfs.DataNode\$PacketResponder
EVENT TEMPLATE	Received block <*> of size <*> from /<*>
PARAMETERS	["blk_-562725280853087685", "67108864", "10.251.91.84"]

Explainability

Generate Human-readable Titles for Log Anomaly

 Log anomaly - anomaly-detector

Abnormal behavior found in microservice: ts-route-mongo



Title:

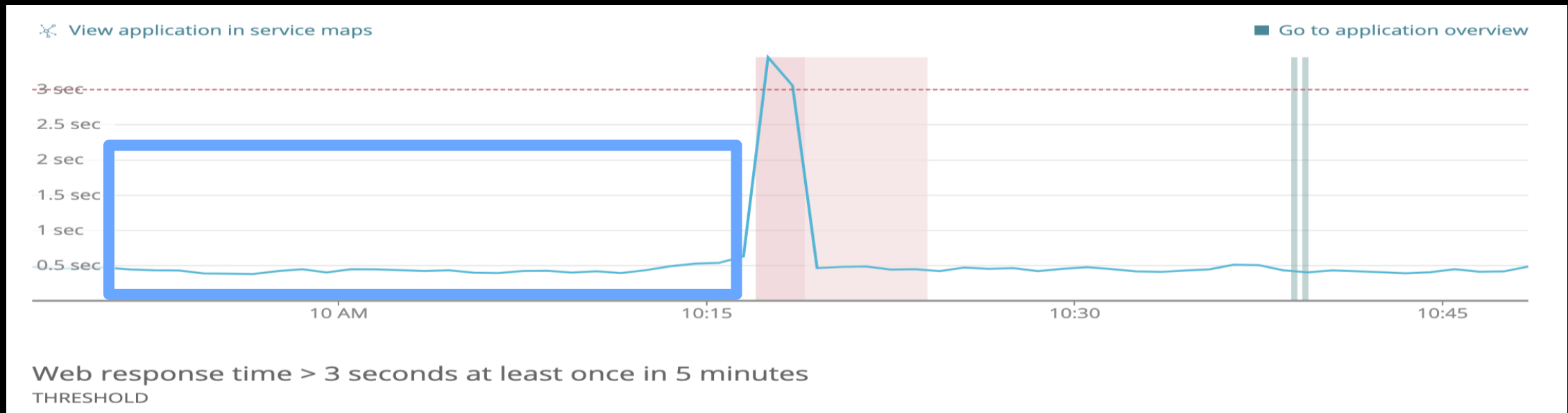
- Abnormal behavior found in component *ts-assurance-mongo*
- 4 errors found in pod *ts-assurance-mongo-54b557bfbc*

```
NETWORK [listener] connection accepted from <*> #<*> (<*>
connections now open): 2
NETWORK <*> end connection <*> (<*> connections now open): 4
```

Allow SREs to drill down into log template and its messages for individual log anomalies

Adjust Severity Level and Grouping based on Causality Scores

Client Engagement

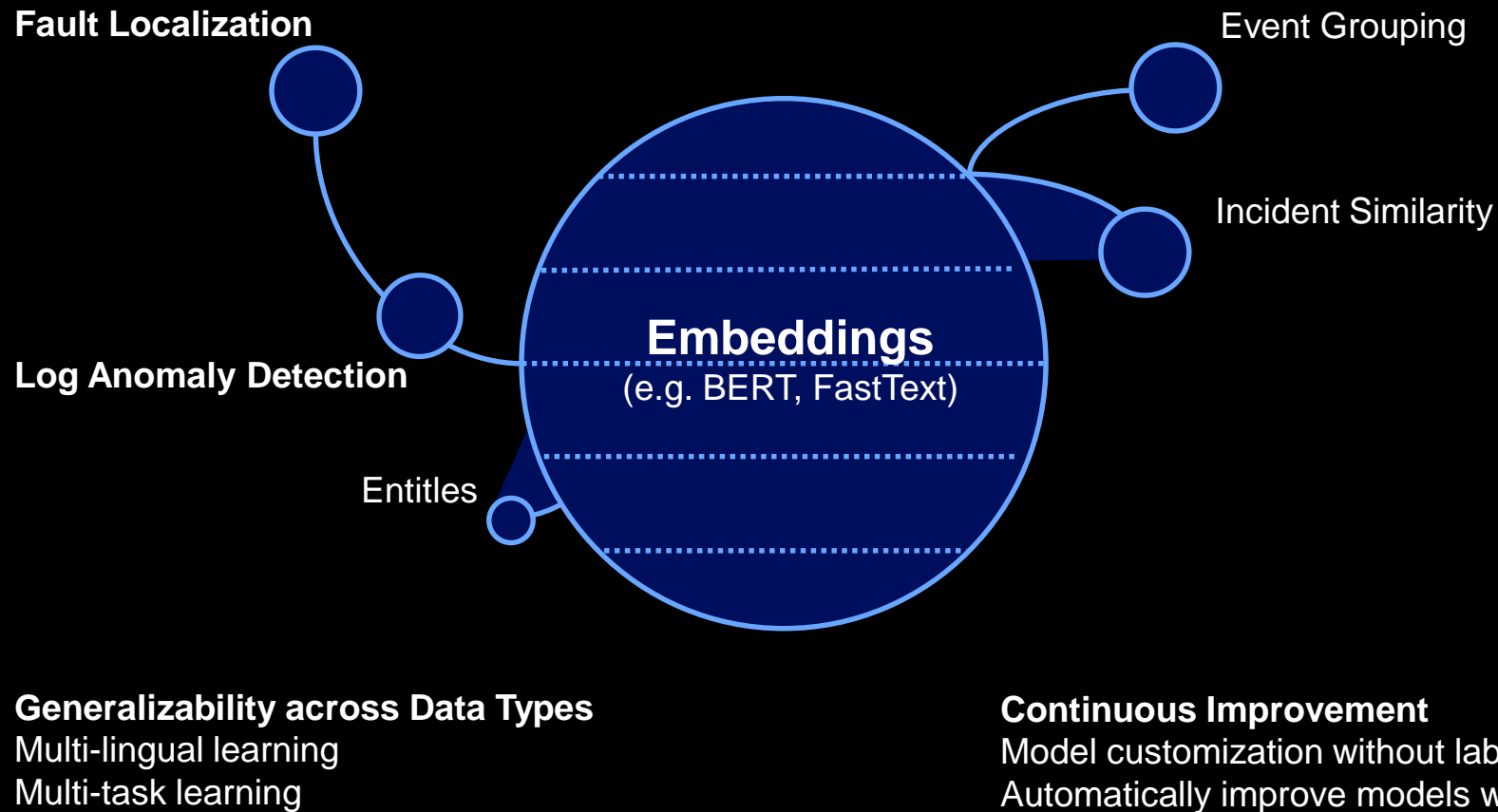


Telcom Company: *"The results (log anomaly results) from my perspective were absolutely amazing, ... (log anomaly results) have provided from near realtime reactive + 6.5hrs predictive. I do not think the client could ask for anything better than these."*

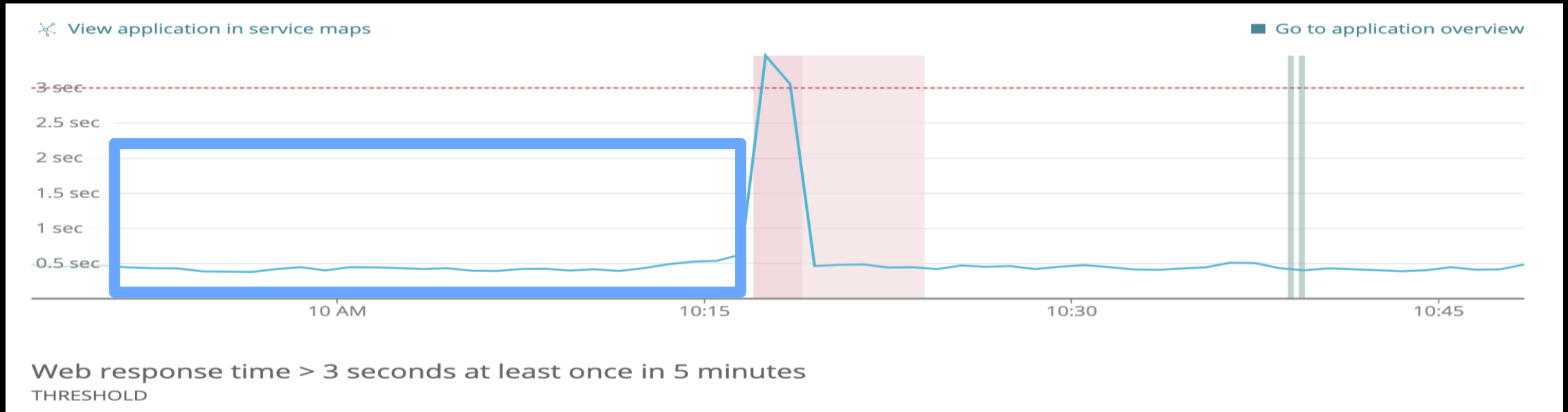
Marketing Company: *"... using when Watson AIOps, we were able to find and localize new anomalies within our data sets which helped our teams address potential long-standing issues and has improved our overall service levels and incident responses."*

Next Step - Uses Language Models to Optimize IT Operations Management

One-stop solution to encode text for IT Operations Management Tasks



Next Step – Golden Signal Driven Log Anomaly Detection & Fault Localization



Model for multiple components (Detection & Fault Localization)

- Consider golden signals (e.g. Latency).
- Consider trace logs based on discriminatory entities such as "transaction_id".

Out-of-the-box models

- Provide Content Pack, so AIOps can benefit clients on Day 0.

Quality Evaluation

Dataset		Per-class Accuracy	
Data Source	Env.	Normal	Abnormal
Chatbot Service	Cloud	93.3%	66.7%
HDFS	Cloud	99.99%	54.8%
Trian Ticket App	Cloud	95.1%	55.1%
Financial Service	Traditional	99.99%	100%
<XXX> Bank	Traditional	91%	60%
<YYY> Bank	Traditional	96%	100%
Marketing Company	Cloud	Was able to detect the client's incident from logs	
Telecom Company	Traditional	Was able to detect the client's incident from logs	

IBM Watson AIOps

Log Anomaly Detection

Leverage Watson AIOps to better predict IT outages from logs.

Anbang Xu, anbangxu@us.ibm.com

Rama Akkiraju, akkiraju@us.ibm.com

