Extending Dynatrace

Legal Disclaimer

This presentation includes certain "forward-looking statements" within the meaning of the Private Securities Litigation Reform Act of 1995, including statements regarding management's expectations of future financial and operational performance and operational expenditures, expected growth, and business outlook, including our financial guidance for the third quarter and full year of fiscal 2021, statements regarding the impact of the COVID-19 pandemic and related economic conditions on our business and results of operations; and statements regarding the size of our market and our positioning for capturing a larger share of our market. These forward-looking statements include, but are not limited to, plans, objectives, expectations and intentions and other statements contained in this press release that are not historical facts and statements identified by words such as "expects," "anticipates," "intends," "plans," "believes," "seeks," "estimates" or words of similar meaning. These forward-looking statements reflect our current views about our plans, intentions, expectations, strategies and prospects, which are based on the information currently available to us and on assumptions we have made. Although we believe that our plans, intentions, expectations, strategies and prospects as reflected in or suggested by those forward-looking statements are reasonable, we can give no assurance that the plans, intentions, expectations or strategies will be attained or achieved. Furthermore, actual results may differ materially from those described in the forward-looking statements and will be affected by a variety of risks and factors that are beyond our control including, without limitation, the effect of the COVID-19 pandemic on our business operations and demand for our products as well as its impact on general economic and financial market conditions; our ability to maintain our subscription revenue growth rates in future periods, our ability to service our substantial level of indebtedness, market adoption of software intelligence solutions for application performance monitoring, digital experience monitoring and infrastructure monitoring, continued spending on and demand for software intelligence solutions, our ability to maintain and acquire new customers, our ability to differentiate our platform from competing products and technologies; our ability to successfully recruit and retain highly-qualified personnel; the price volatility of our common stock, and other risks set forth under the caption "Risk Factors" in our Quarterly Report on Form 10-Q filed on [February 3, 2021], and our other SEC filings. We assume no obligation to update any forward-looking statements contained in this document as a result of new information, future events or otherwise.

This presentation also contains estimates and other statistical data made by independent parties and by the Company relating to market size and growth and other data about the Company's industry. This data involves a number of assumptions and limitations, and you are cautioned not to give undue weight to such estimates. Neither the Company nor any other person makes any representation as to the accuracy or completeness of such data or undertakes any obligation to update such data after the date of this presentation. In addition, projections, assumptions and estimates of the Company's future performance and the future performance of the markets in which the Company operates are necessarily subject to a high degree of uncertainty and risk. By attending or receiving this presentation you acknowledge that you will be solely responsible for your own assessment of the market and the Company's market position and that you will conduct your own analysis and be solely responsible for forming your own view of the potential future performance of the Company's business.

This presentation includes non-GAAP financial measures which have certain limitations and should not be considered in isolation, or as alternatives to or substitutes for, financial measures determined in accordance with generally accepted accounting principles in the United States ("GAAP"). The non-GAAP measures as defined by the Company may not be comparable to similar non-GAAP measures presented by other companies. The Company's presentation of such measures, which may include adjustments to exclude unusual or non-recurring items, should not be construed as an inference that its future results will be unaffected by these or other unusual or non-recurring items. A reconciliation of these non-GAAP financial measures to the most directly comparable GAAP financial measures is included in the Appendix to these slides.

© 2021 Dynatrace, Inc. All Rights Reserved

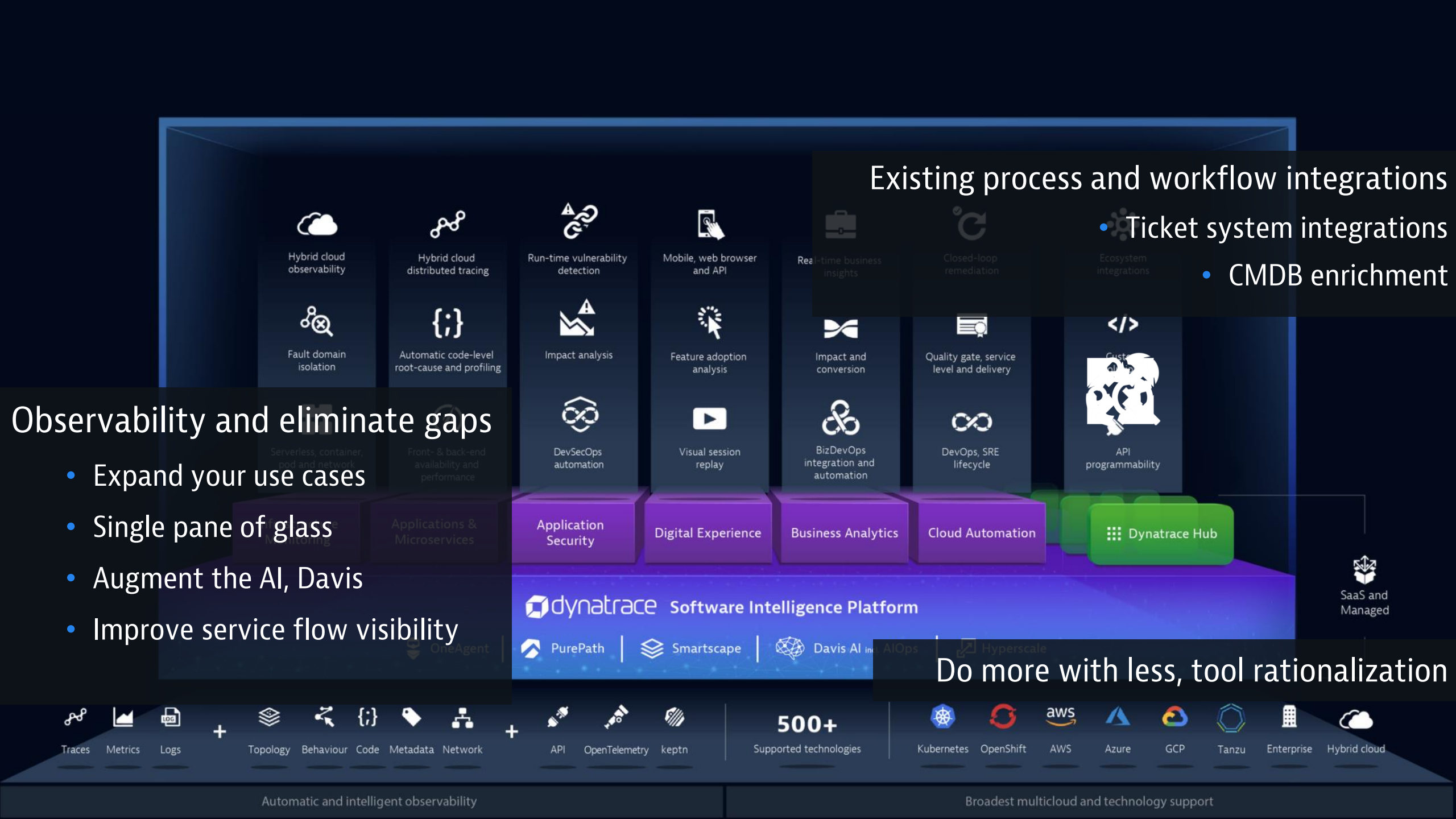
Agenda

- Prepare your environment
- Use cases and methods of extending the platform
- ActiveGate extension framework
- Basic troubleshooting
- Hands on: Deploy your first extension
- Break
- Advanced troubleshooting
- Hands on: Create your first extension
- Best practices
- Roadmap
- Hands on: Extensions framework 2.0

Install OneAgent and ActiveGate

While this session will focus on ActiveGate extensions, which does not require the OneAgent to be installed, it is required for one of the hands-ons.

1. Connect to your Windows instance using remote desktop
2. Execute the “RunMe.bat” file on the desktop
3. Log in to your Dynatrace SaaS instance using Chrome or Firefox
4. Navigate to “Deploy Dynatrace”
5. Download and install the OneAgent
6. Download and install the default ActiveGate



Existing process and workflow integrations

- Ticket system integrations
- CMDB enrichment

Observability and eliminate gaps

- Expand your use cases
- Single pane of glass
- Augment the AI, Davis
- Improve service flow visibility

Do more with less, tool rationalization

What have other customers done?

Observability and eliminate gaps

- Single pane of glass into SAP Hana database replicati on – Italian energy company
- Proactive monitoring of core infrastructure components and certificates – US government department

Existing process and workflow integrations

- Ticket system integrations
- CMDB enrichment

Observability and eliminate gaps

- Expand your use cases
- Single pane of glass
- Augment the AI, Davis
- Improve service flow visibility

alization

y scripts to AI powered answers – US pharmaceutical industry

a database for business decisions based on outages – US energy company

/ integrations

DB integration to make the cor

Do more with less, tool rationalization

- Integrate Dynatrace problems with Remedy with ticket routing information – German bank

Dynatrace has you covered

We walk the talk – More than 50 out of the box extensions, our official ServiceNow plugin and autonomous cloud offerings are built on top of platform extensibility

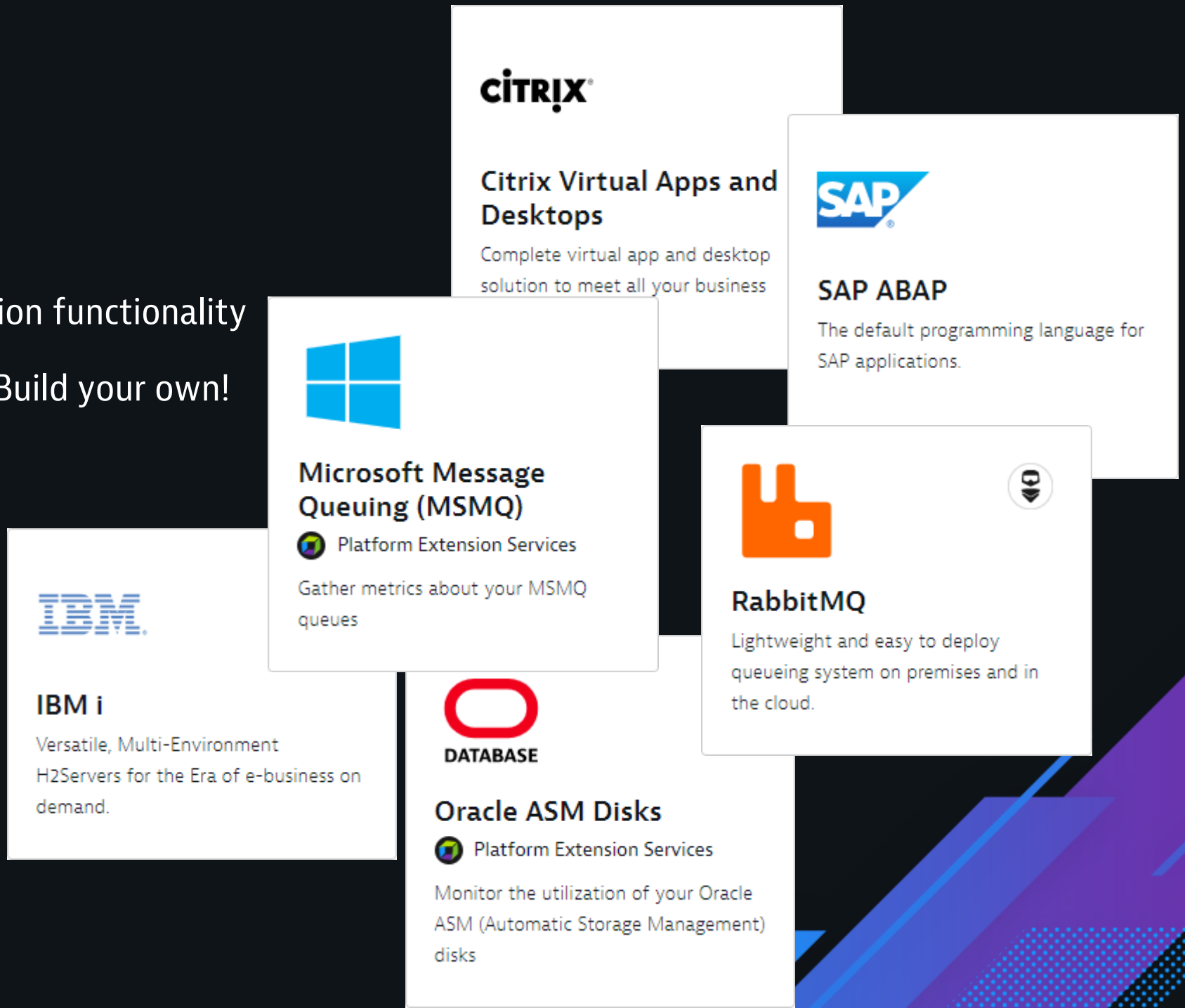
Dynatrace services has a team of global extension architects dedicated to making you successful. Example offerings with an average effort:

- Assessment - What makes sense to bring into Dynatrace and how to get the data (1-2 days)
- Workshop - Overview training, best practices and start working on your requirements together to build a good foundation (4-8 hours)
- Implementation - Perform all or a part of an implementation. Tasks might include design, development, documentation and handover. The team has the experience of more than 150 delivered extensions (2-10 days, dependent on the data source and complexity)

Prefer to do it yourself? Visit developer.dynatrace.com to sign up for a developer instance and connect with the community of more than 300 customers and partners

Dynatrace Hub

- 500+ technology tiles
- Many are built using extension functionality
- Don't find what you need? Build your own!



Send data to Dynatrace

- Metrics
- Events
- PurePaths
- User Sessions
- Synthetic tests

	Metrics	Events	Requires OneAgent	Requires ActiveGate	PurePaths	User Sessions	Synthetic tests
OneAgent extension	✓	✓	✓	✗	✗	✗	✗
ActiveGate extension	✓	✓	✗	✓	✗	✗	✗
API	✓	✓	✗	✗	✗	✗	✓
OneAgent SDK	✓	✗	✓	✗	✓	✗	✗
OpenKit	✗	✗	✗	✗	✗	✓	✗

Extension Methods

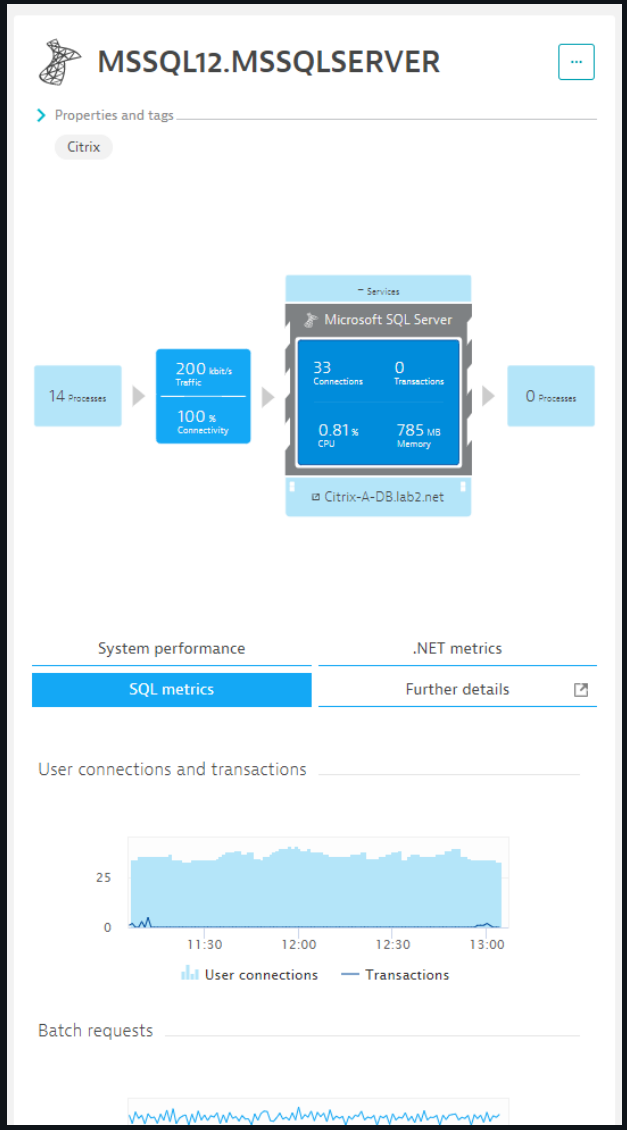
Overview

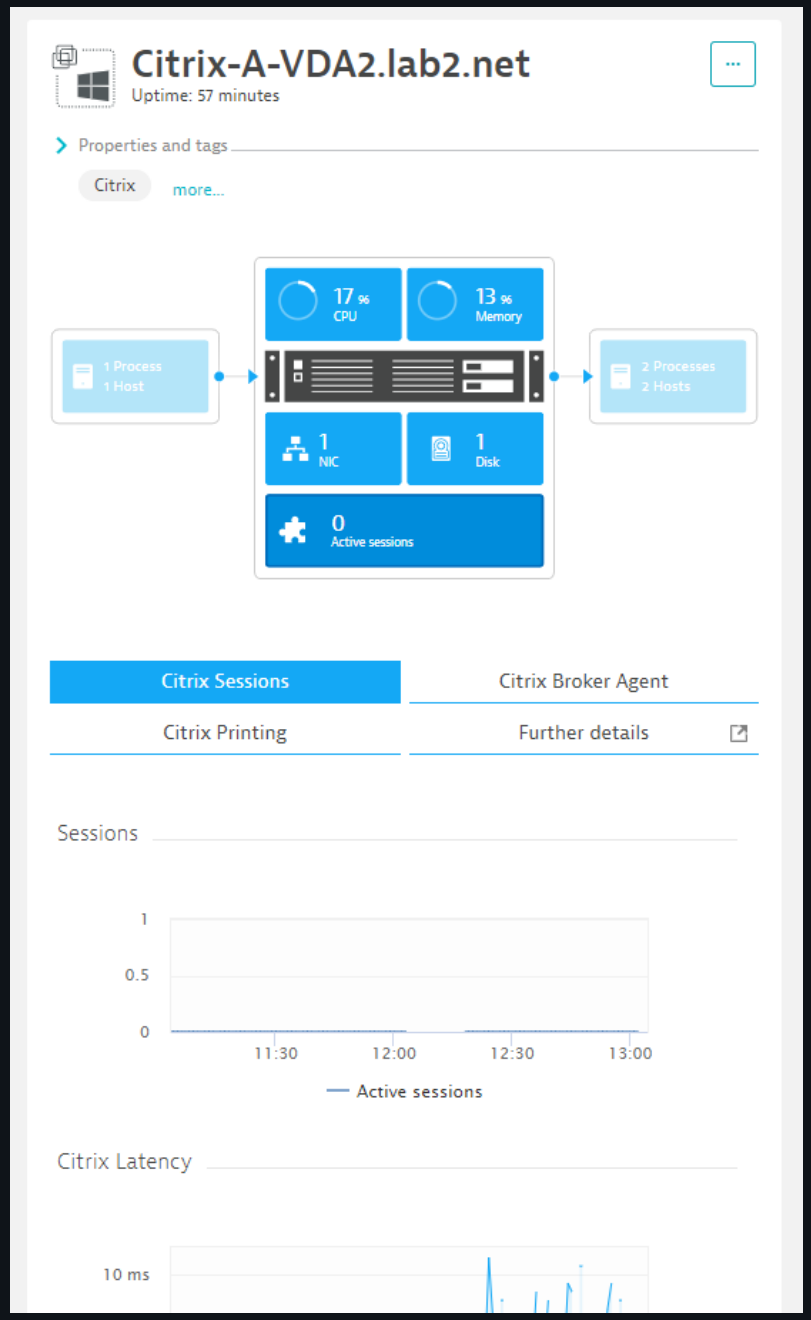
OneAgent Extension

ActiveGate Extension

Other methods

OneAgent Extension





OneAgent Extension

- Capabilities
  - Enhance processes running on your host
  - Extend the Dynatrace host monitoring
  - Add metrics, events, properties, and charts
  - Python or JMX/PMI
- Examples
  - Redis
  - RabbitMQ
  - Tibco EMS
  - Siebel
  - Connection pools
- Don't use if
  - You cannot run OneAgent Extensions
  - You want to capture PurePaths or user sessions

Extension Methods

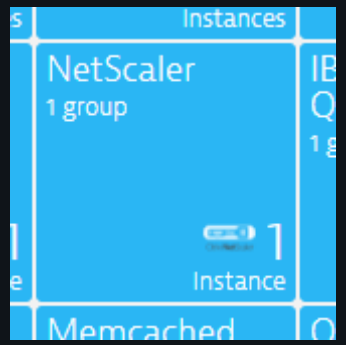
Overview

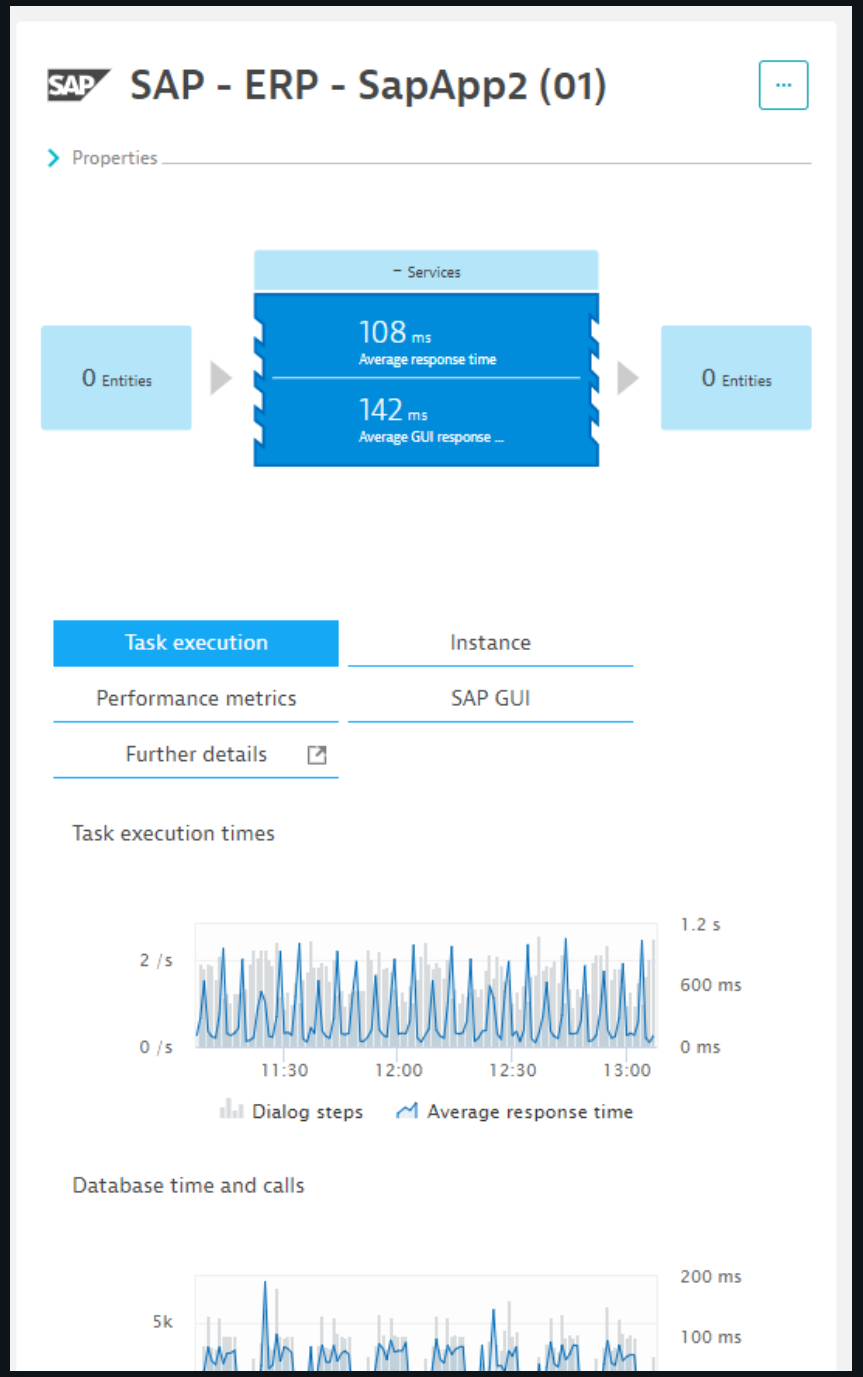
OneAgent Extensions

ActiveGate Extension

Other methods

ActiveGate Extension





ActiveGate Extension

- Capabilities
  - Monitor components where you can't run OneAgent Extensions
  - Add metrics, events, properties, charts, IPs, and ports
  - Create Custom Devices (or use it as a scheduler!)
- Examples
  - f5
  - DataPower
  - NetScaler
  - Juniper
  - IBM i
- Don't use if
  - The processes are already monitored by the OneAgent
  - There is no external interface to get metrics/events

Extension Methods

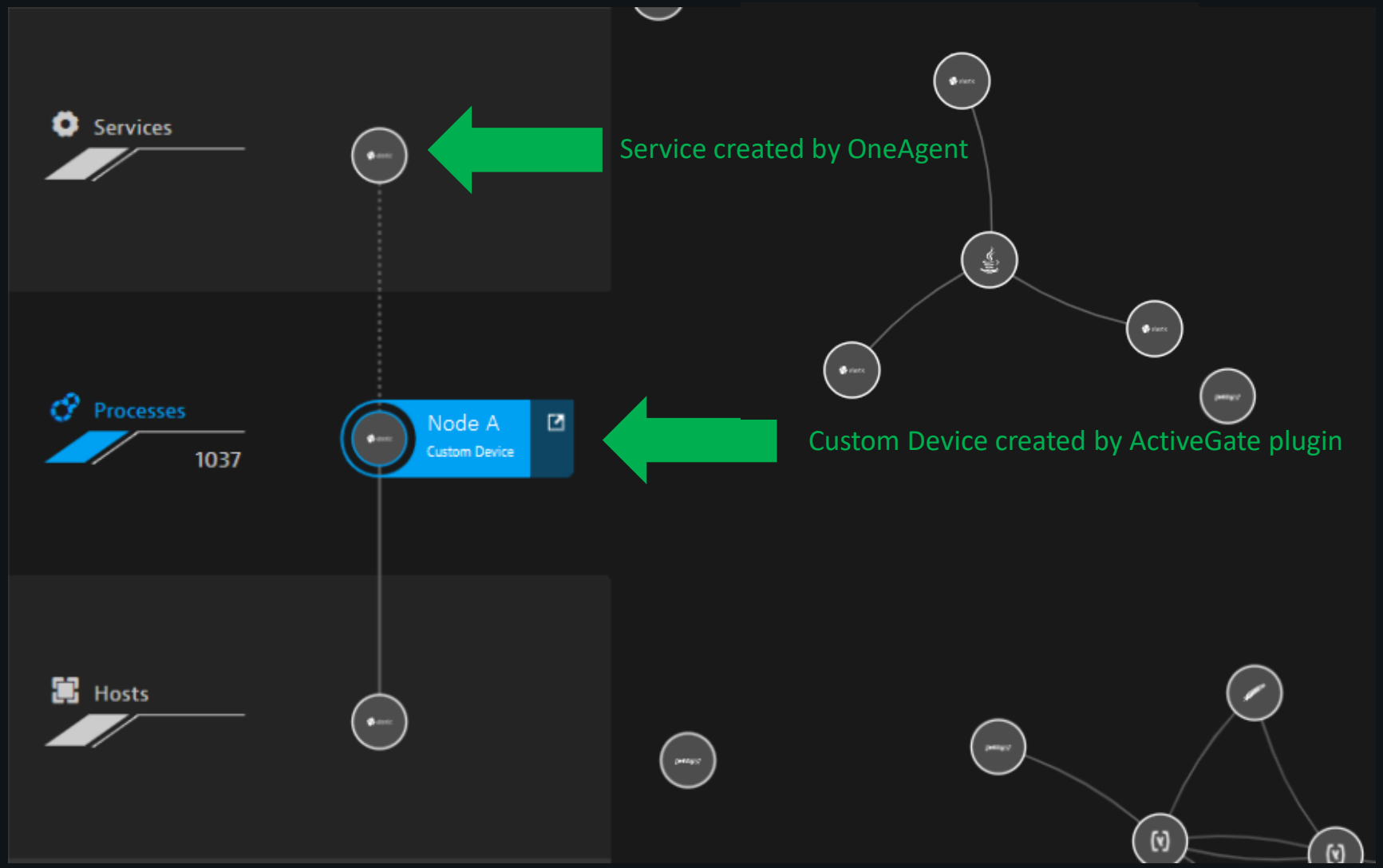
Overview

OneAgent Extensions

ActiveGate Extension

Other methods

Data gathered by Extensions is integrated and analyzed by the AI



Extension Methods

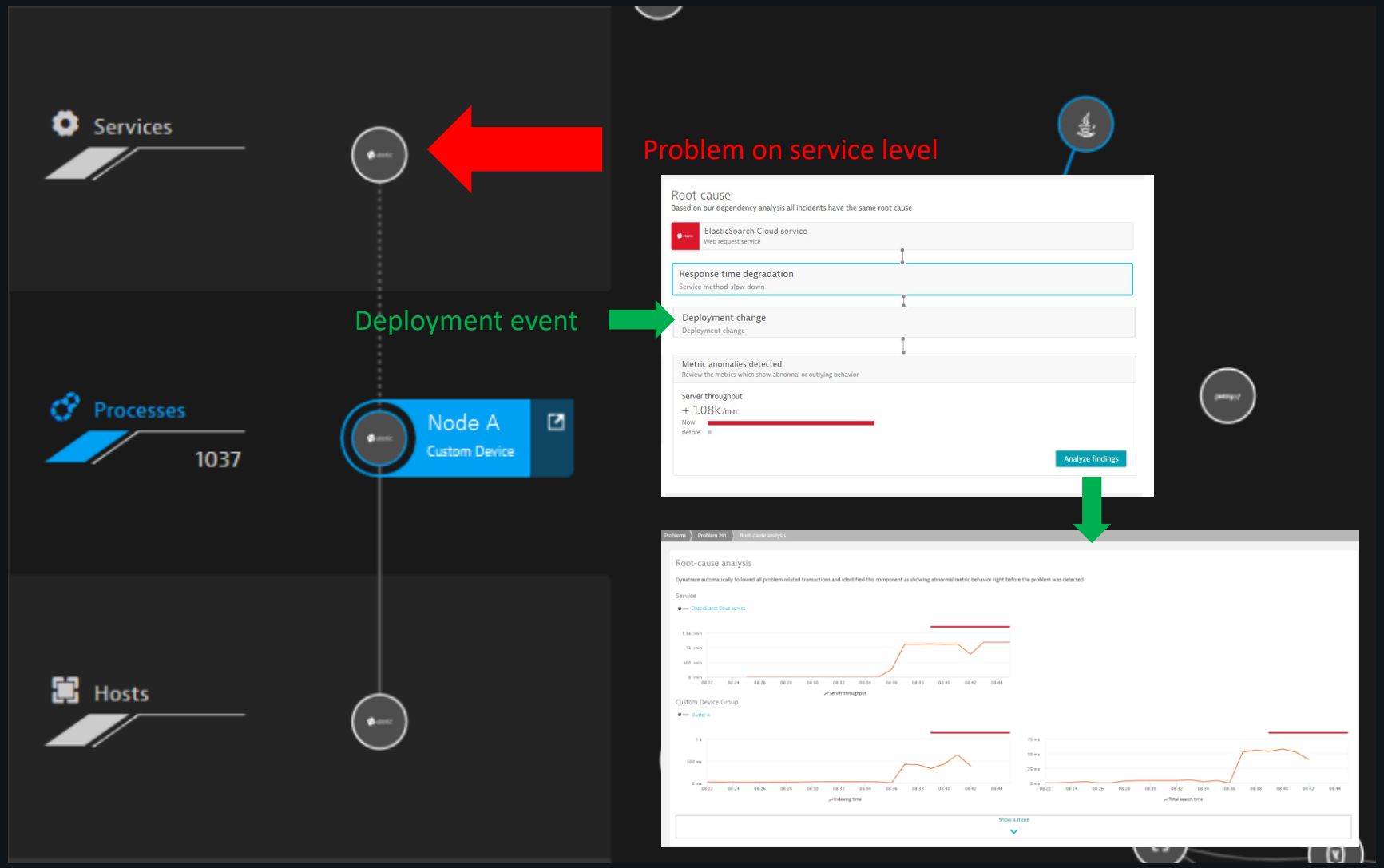
Overview

OneAgent Extensions

ActiveGate Extension

Other methods

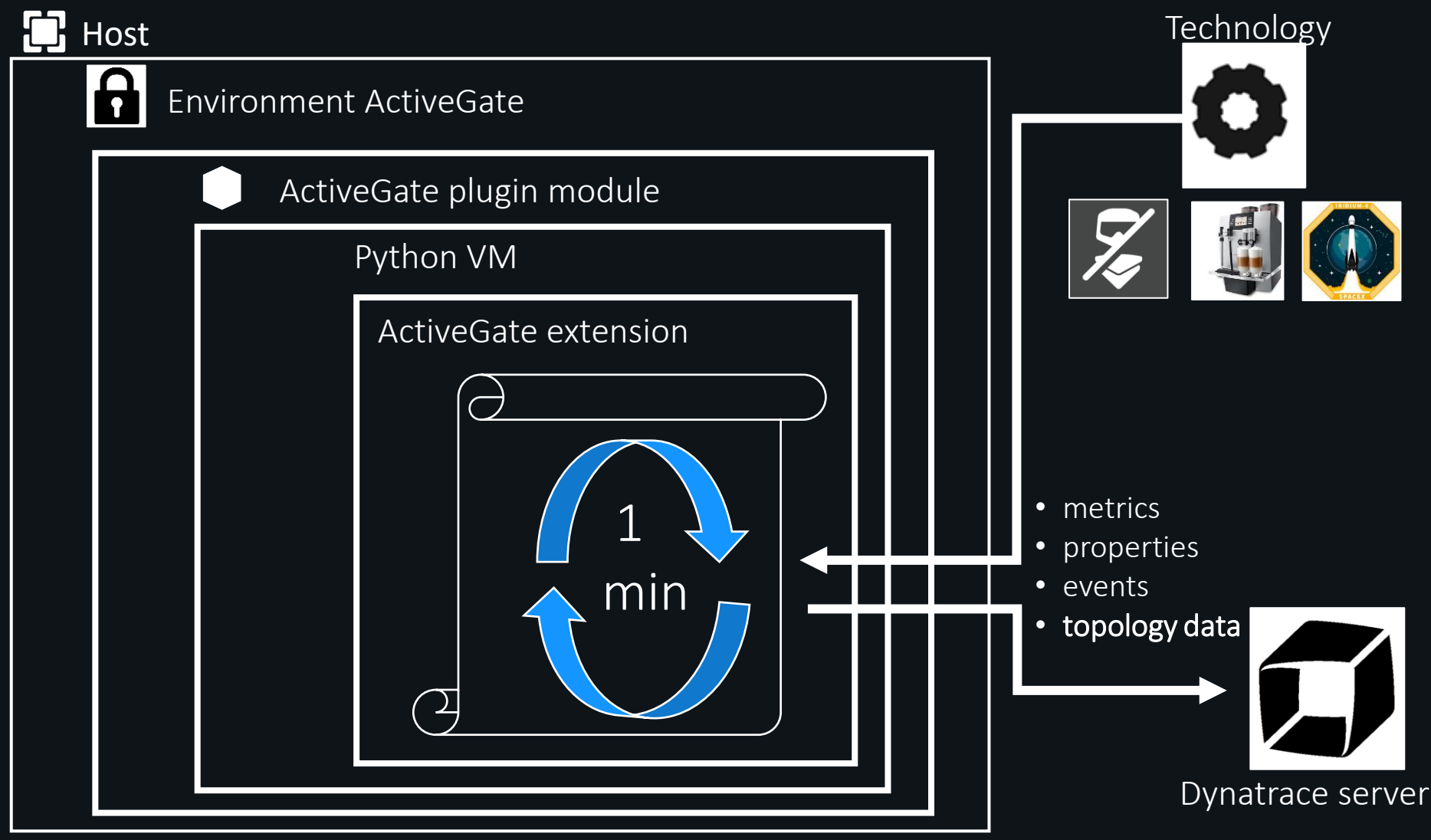
AI 2.0 detects root causes in plugin metrics without thresholds



Other methods

For additional details and the use cases of other methods, join the breakout session **“Opening Dynatrace to your ecosystem: what you need to know about Dynatrace APIs, SDKs and Extensions”**

The ActiveGate extensions framework architecture



Extension Framework

- JSON file and Python code
- ActiveGate and OneAgent Extensions
- Dynatrace server only cares about JSON file
- During development
  - Python 3.6
  - properties.json
  - snapshot_simulator.json (OneAgent extension)
  - plugin_sdk library
- To deploy:
  - Extension zip file

Plugin.json

- JSON file that defines the extension
- Mandatory:
  - name
  - metricGroup
  - version
  - type
  - source
  - metrics
- Optional:
  - ui
  - ConfigUI
  - properties
- Reference
  - https://dynatrace.github.io/plugin-sdk/api/plugin_json_apidoc.html

```
{
    "name": "custom.remote.python.juniper",
    "metricGroup": "tech.Juniper",
    "version": "1.086",
    "type": "python",
    "entity": "CUSTOM_DEVICE",
    "technologies": ["Juniper"],
    "favicon": "https://imgur.com/H0kgC2m.png",
    "source": {
        "package": "custom_juniper_plugin_remote",
        "className": "CustomJuniperPluginRemote",
        "install_requires": ["junos-eznc==2.2.0"],
        "activation": "Remote"
    },
    "properties": [{
            "key": "device_address",
            "type": "String"
        },
        {
            "key": "device_port",
            "type": "String"
        },
```

Python API

- Plugin SDK Python Library
  - Develop and simulate extensions
  - Send metrics and events, set properties
- Plugins extend either:
  - RemoteBasePlugin (ActiveGate extensions)
  - BasePlugin (OneAgent plugins)
- And must implement:
  - query (called every minute)

```python
class CustomDBQueryPluginRemote(RemoteBasePlugin):
    def initialize(self, **kwargs):
        self.execution_counter = 0

    def query(self, **kwargs):
        self.config = kwargs["config"]

        device_name = f'{self.config["database_host"]}:{self.config["database_port"]}'
        if "alias" in self.config and self.config["alias"]:
            device_name = self.config["alias"]
```

- Optionally:
  - Initialize (called at startup and endpoint update)

HOT – Deploy

Deploy

Visualize Data

Deploy Demo Plugin

- We need:
  - The Plugin SDK
  - An API token to upload extensions
- Plugin SDK
  1. Navigate to your Dynatrace Tenant
  2. Settings > Monitored technologies > Add new technology monitoring > Add ActiveGate Plugin > Download Plugin SDK
- Token
  1. Settings > Integration > Dynatrace API
  2. Create a token with the following permission scopes:
     1. Read+Write configuration
     2. Read+Write extensions
     3. Read+write extension environment configurations
  3. **Copy the API token into a file**

HOT – Deploy

Deploy

Visualize Data

Deploy Demo Extension

1. RDP to your ActiveGate server
2. Configure Token
   - There are three options
     1. Create the file plugin_upload.token at C:\Program Files\dynatrace\remotepluginmodule\agent\conf\; OR
     2. Set the **ONEAGENT_PLUGIN_UPLOAD_TOKEN** environment variable, pointing to a **plugin_upload.token** file; OR
     3. Pass the token later, when we execute the oneagent_build_plugin command
3. Install Plugin SDK
   1. Unzip the SDK zip file
   2. Open Visual Studio Code as Administrator
   3. Open the demo application plugin directory
      1. File > Open Folder
         - <SDK_Folder>\examples\demo_activegate_plugin
   4. pip install <path_to_plugin_sdk_wheel>

HOT – Deploy

Deploy

Visualize Data

Deploy Demo Extension

1. Start the demo app web service
   - python -m plugin_sdk.demo_app

```
D:\projects\python\plugin-tools\sdk\185\examples\demo_activegate_plugin>python -m plugin_sdk.demo_app
Bottle v0.12.18 server starting up (using WSGIRefServer())...
Listening on http://0.0.0.0:8769/
Hit Ctrl-C to quit.
```

2. Simulate your extension
   - oneagent_sim

3. Deploy
   - oneagent_build_plugin

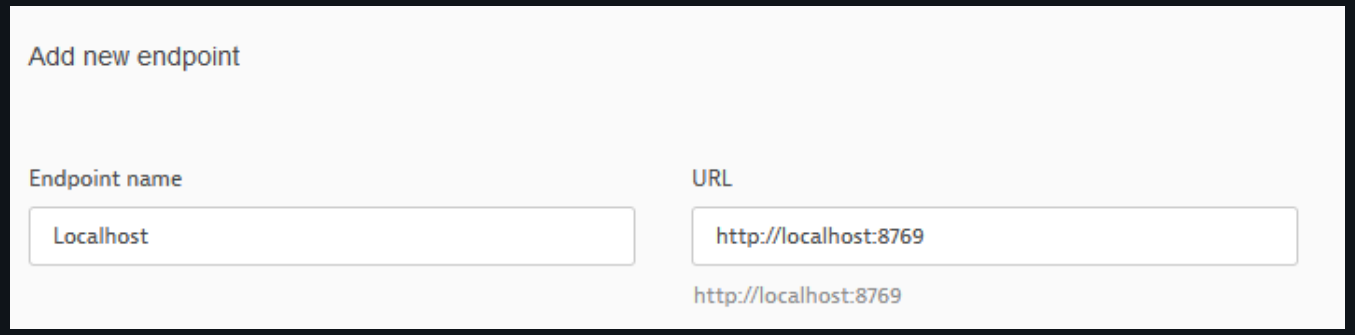
HOT – Deploy

Deploy the demo plugin

Visualize Data

Deploy Demo Extension

- Make sure the extension is uploaded
  - Settings > Monitored technologies > Custom extensions > ActiveGate Demo Plugin
- Add an endpoint



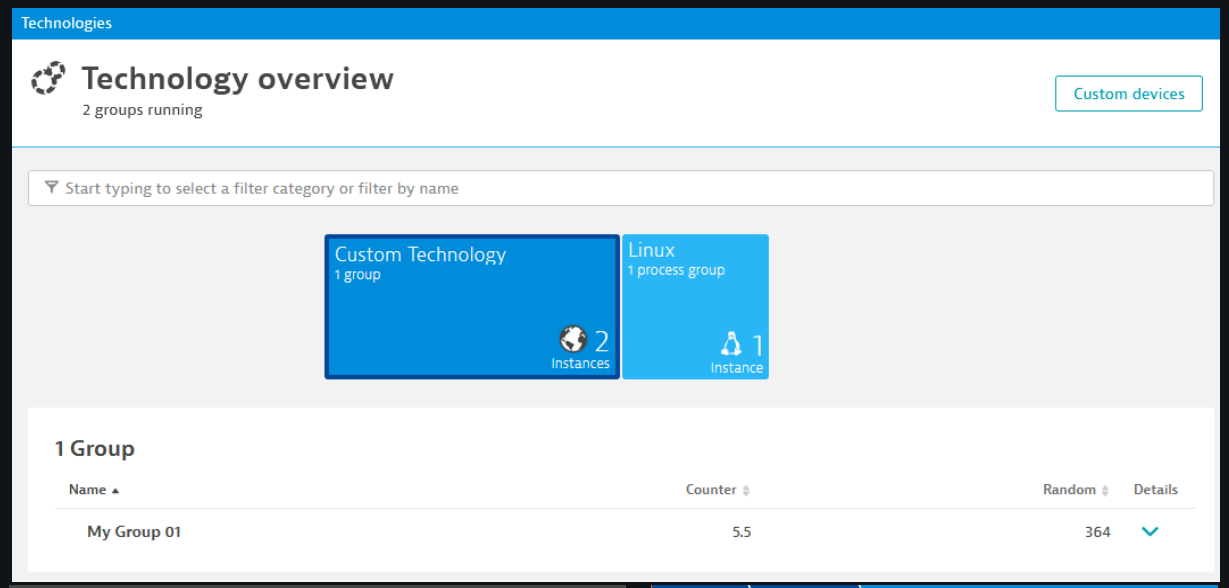
HOT – Deploy

Deploy the demo plugin

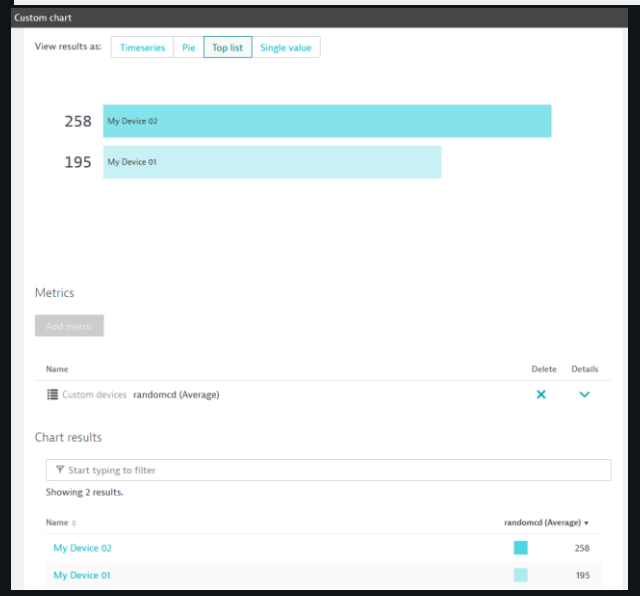
Visualize Data

Deploy Demo Extension

- Technologies



- Custom charts



BREAK

Troubleshooting

- Every extension must:
  - Run in less than a minute
- OneAgent extensions must:
  - Be initialized
    - Process Type found? Singleton?
- Never forget to deploy
  - Extensions must be deployed to the hosts
- Uncaught exceptions will put the extension in an error state
  - If it happens frequently, the extension won't run again
- Too much resource usage will disable your extension
  - Look for throttle messages in the log

Troubleshooting

- ActiveGate (RemotePluginModule)
  - Linux: /var/lib/dynatrace/remotepluginmodule/log/remotepluginmodule
  - Windows: C:\ProgramData\dynatrace\remotepluginmodule\log\remoteplugin
- OneAgent
  - Linux: /opt/dynatrace/oneagent/log/plugin
  - Windows: C:\ProgramData\dynatrace\oneagent\log\plugin
- Important files:
  - ruxitagent_remotepluginagent_<PID>.0.log
  - The extension log, under the plugin_id folder (ActiveGate)
- Remember to write good log entries!
  - log.exception()

Troubleshooting

Basics

Logs

Advanced

Advanced Troubleshooting

- sys.path matters
  - Don't use common names for files, like utils.py, common.py
- Look out for compiled libraries (.so and .pyd files)
  - Build for Linux on Linux, and Windows on Windows (or combine it yourself)
- Some libraries already come with the RPM and OneAgent:
  - requests
  - winpdh
  - Do not rely on it!
- Python is embedded, 3.6
- Build times out at 60 seconds
  - You can work around this if necessary

C:\Program Files\Python36\Lib\site-packages\plugin_sdk\plugin_build.py - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

plugin_build.py

```
247 def _install_plugin_wheel(tmp_installpath, whlpath):
248     import subprocess
249
250     try:
251         subprocess.check_call(
252             [
253                 sys.executable,
254                 "-m",
255                 "pip",
256                 "install",
257                 "--ignore-installed",
258                 "--only-binary=:none:",
259                 "--disable-pip-version-check",
260                 "--target=" + tmp_installpath,
261                 "--upgrade",
262                 "--no-compile",
263                 whlpath,
264             ],
265             timeout=60,
266         )
267     except (subprocess.CalledProcessError, subprocess.TimeoutExpired):
268         print("Failed to install the plugin into temporary location")
269         exit(-1)
```

Space X Extension

Topology Builder

Requesting Data

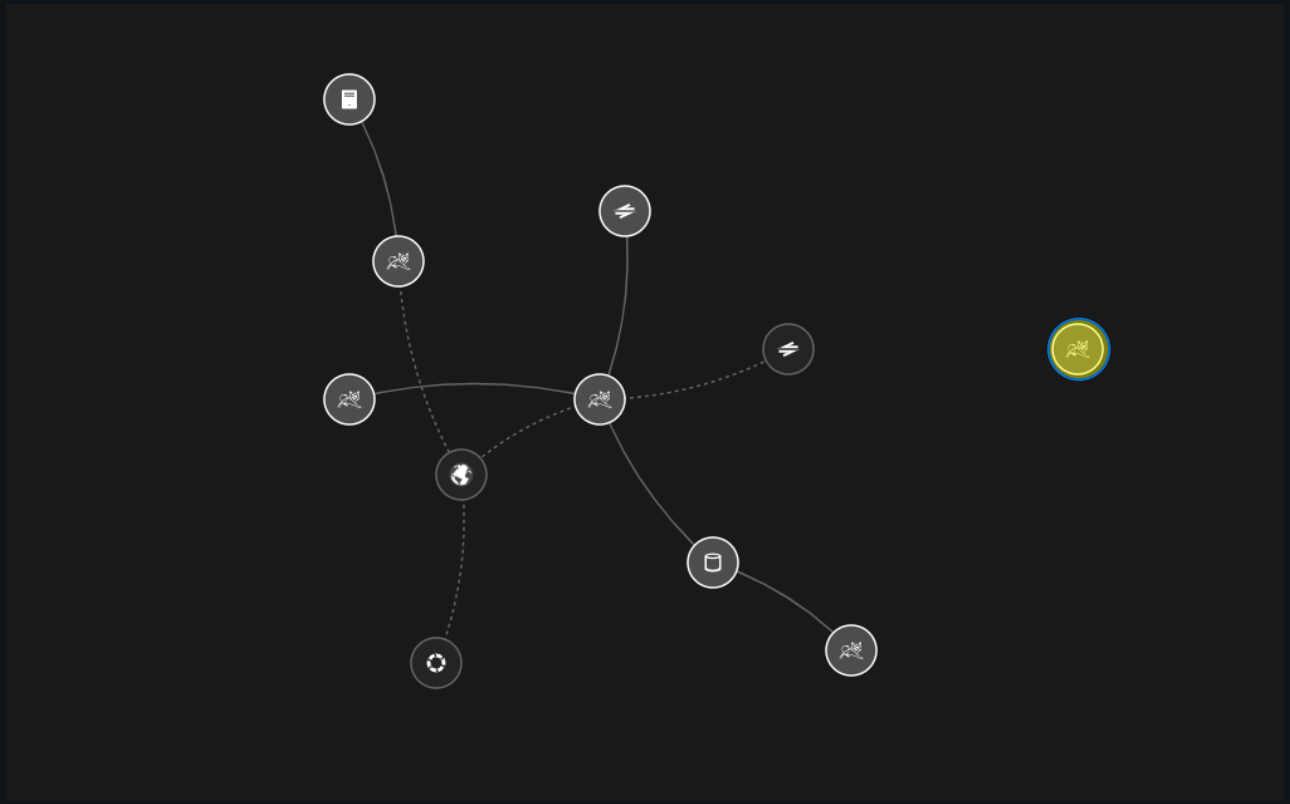
Metrics

Multi Dimensional Metrics

Configuration

Topology Builder – Custom Devices

Custom Device



Space X Extension

Topology Builder

Requesting Data

Metrics

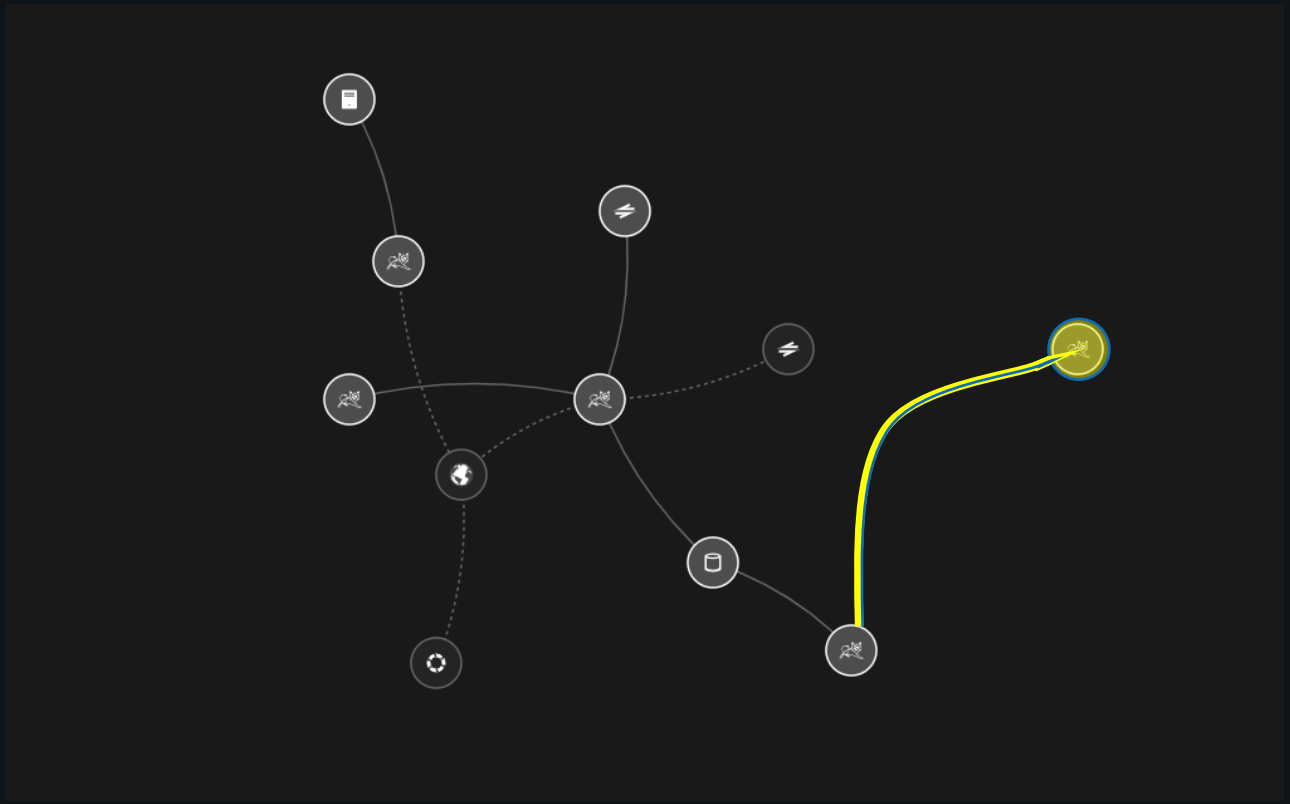
Multi Dimensional Metrics

Configuration

Topology Builder – Custom Devices

Custom Device

Endpoint / IP



Space X Extension

Topology Builder

Requesting Data

Metrics

Multi Dimensional Metrics

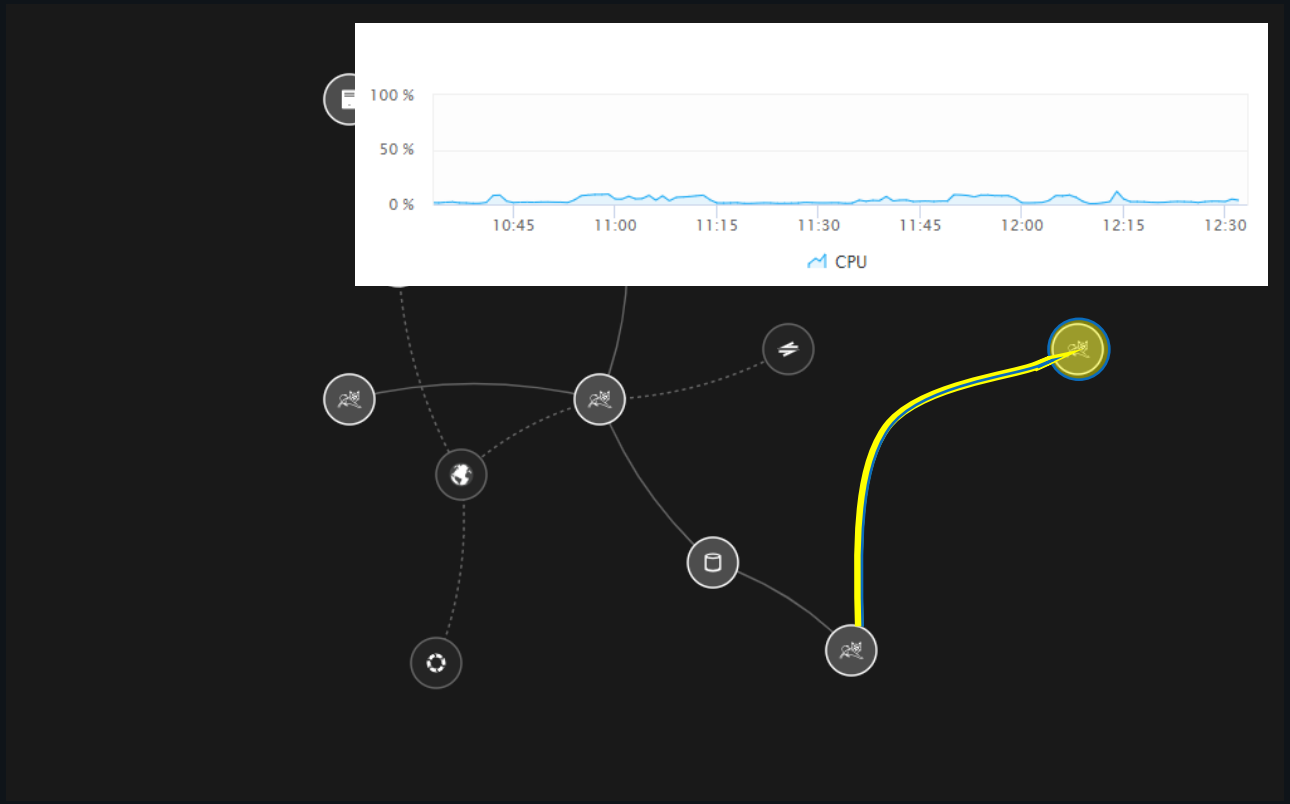
Configuration

Topology Builder – Custom Devices

Custom Device

Endpoint / IP

Metrics



Space X Extension

Topology Builder

Requesting Data

Metrics

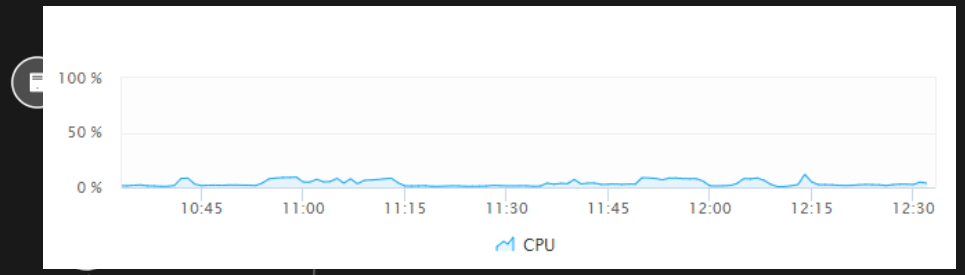
Multi Dimensional Metrics

Configuration

Topology Builder – Custom Devices

Custom Device

- Endpoint / IP
- Metrics
- Properties



TAG009442447542.clients.dynatrace.org

Uptime: 5 hours 50 minutes

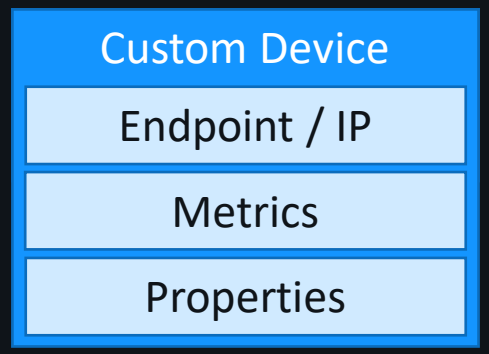
Properties

+ Add tag

Windows 10 Pro 1903, ver. 10.0.18362

Detected name	TAG009442447542.clients.dynatrace.org
OneAgent version	1.181.154.20191211-083443
Architecture	x86, 64-bit
Data center	Linz, Austria
IP addresses	192.168.1.102, 172.31.118.148, 192.168.56.1, and 192.168.99.1
Logical CPU cores	12
Monitoring mode	Full stack
Physical CPU cores	6

Topology Builder – Custom Devices



Space X Extension

Topology Builder

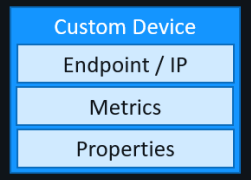
Requesting Data

Metrics

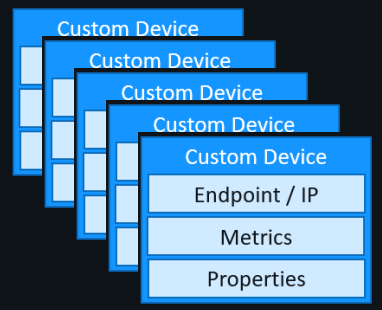
Multi Dimensional Metrics

Configuration

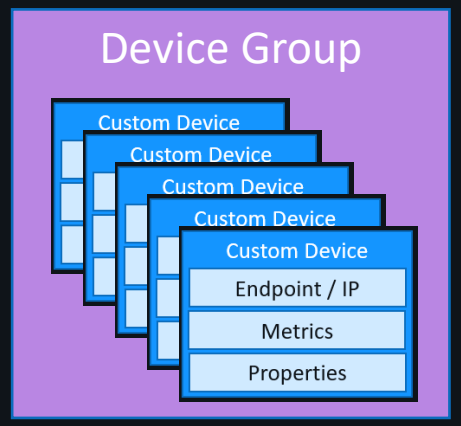
Topology Builder – Custom Devices



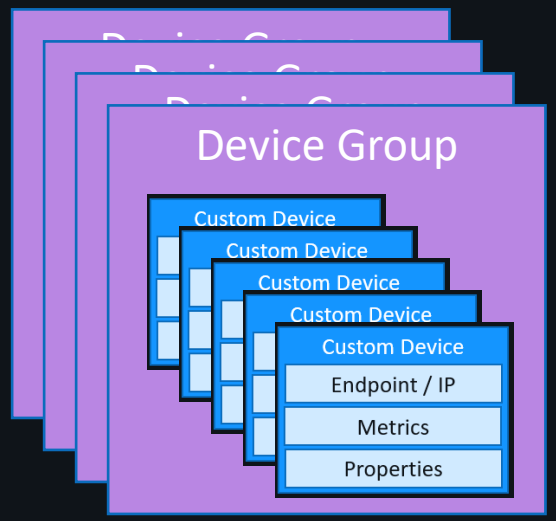
Topology Builder – Custom Devices



Topology Builder – Custom Devices



Topology Builder – Custom Devices



Space X Extension

Topology Builder

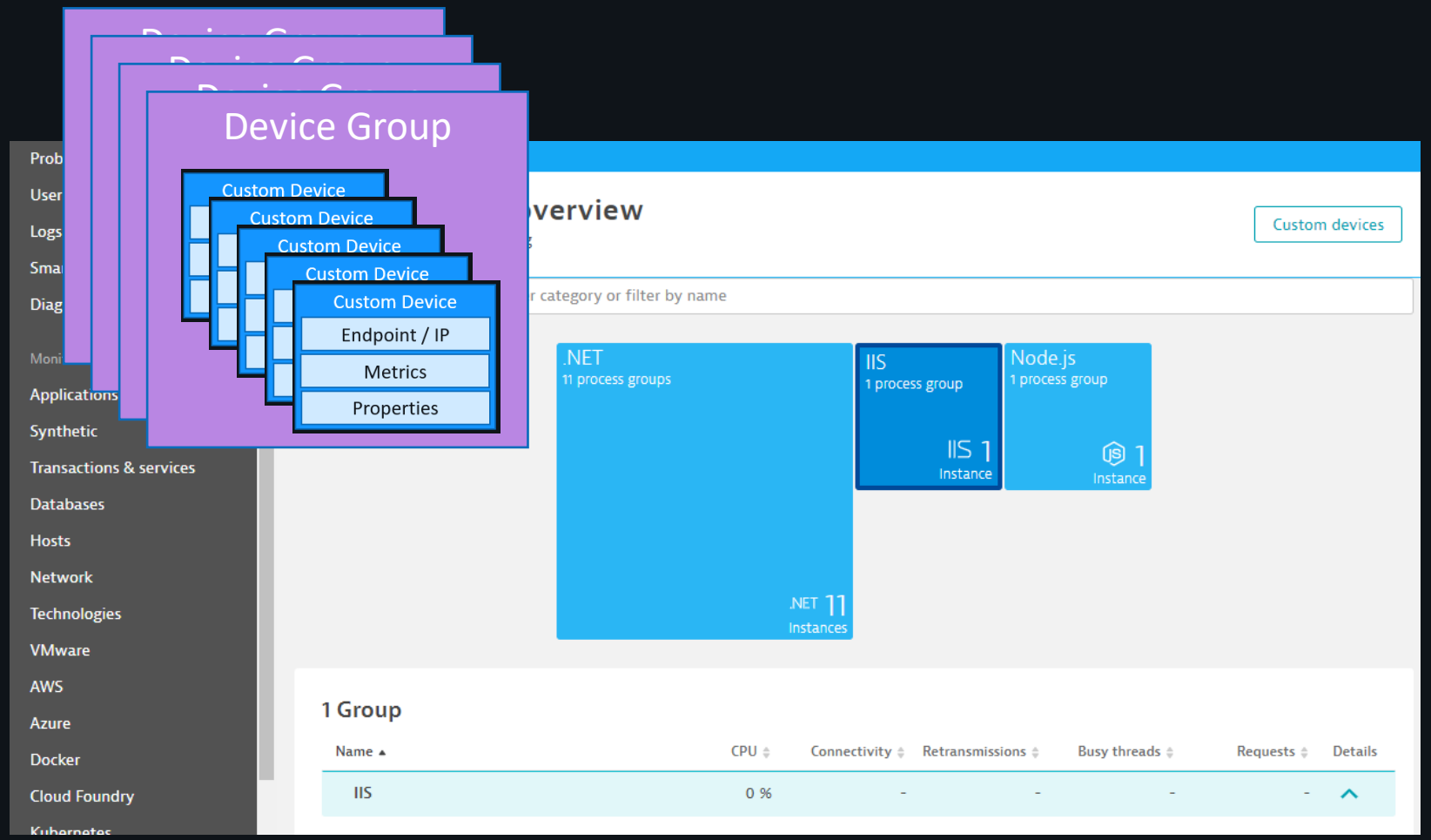
Requesting Data

Metrics

Multi Dimensional Metrics

Configuration

Topology Builder – Custom Devices



Space X Extension

Topology Builder

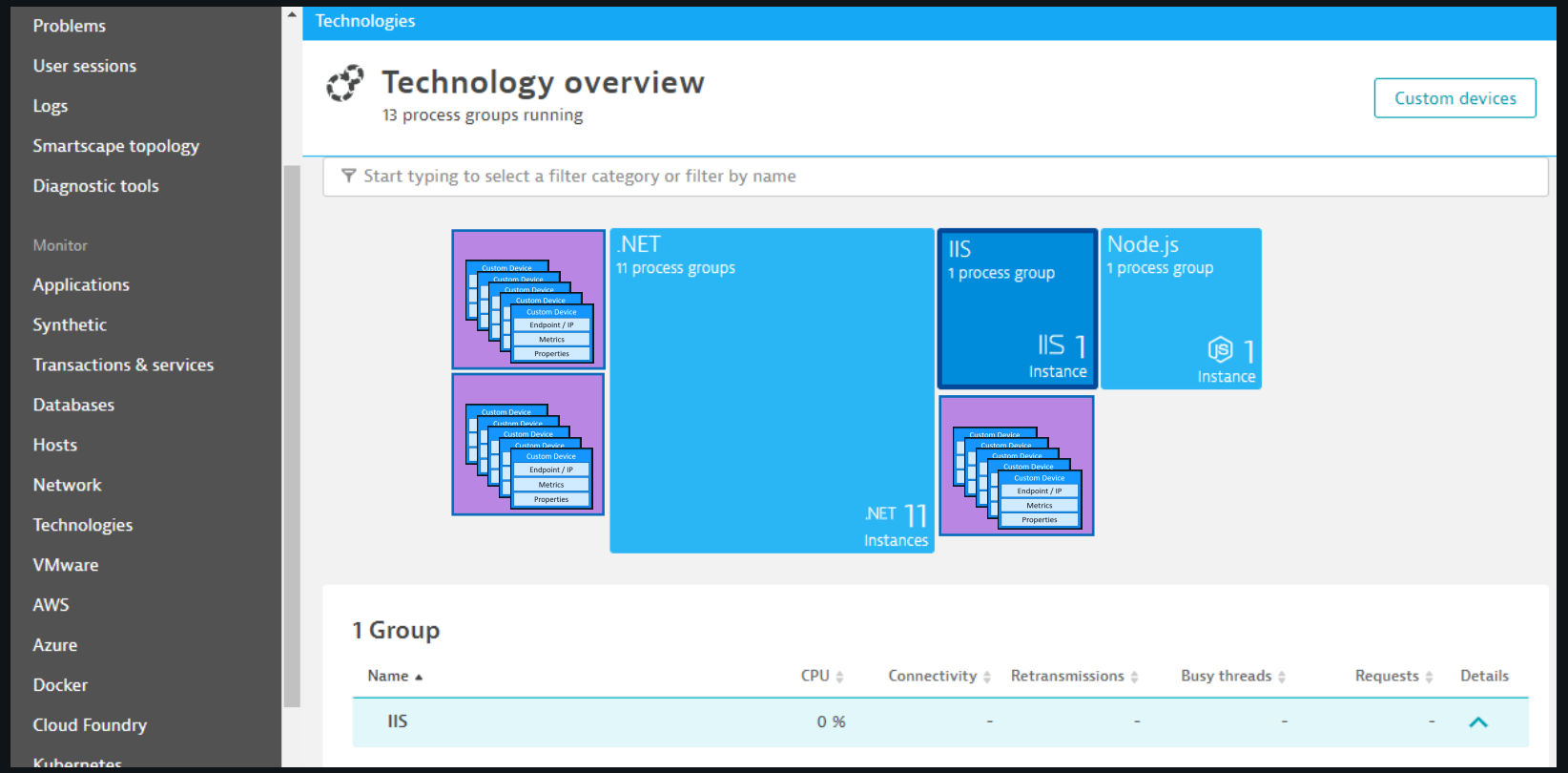
Requesting Data

Metrics

Multi Dimensional Metrics

Configuration

Topology Builder – Custom Devices



Space X Extension

Topology Builder

Requesting Data

Metrics

Multi Dimensional Metrics

Configuration

Requesting Data via HTTP

r/SpaceX API Docs

docs.spacexdata.com

R/SPACEX API DOCS

Introduction

- Capsules
- Cores
- Dragons
- History
- Info
- Landing Pads
- Launches
- Launch Pads
- Missions
- Payloads
- Rockets
- Roadster
- Ships
  - GET All Ships
  - GET One Ship

GET All Ships

https://api.spacexdata.com/v3/ships

Optional Querystrings

Param	Sample	Type	Description
ship_id	MRSTEVEN	string	filter results by ship id
ship_name	MR STEVEN	string	filter results by ship name
ship_model	Marmac 300	string	filter results by ship model
ship_type	Tug	string	filter results by ship type
role	Support Ship	string	filter results by ship role
active	true	boolean	filter results by active ships
imo	7434016	integer	filter results by imo id number
mmsi	367020820	integer	filter results by mmsi id number
abs	571252	integer	filter results by abs id number
class	7604342	integer	filter results by class id number
weight_lbs	588000	integer	filter results by gross weight in lbs

Space X Extension

Topology Builder

Requesting Data

Metrics

Multi Dimensional Metrics

Configuration

Requesting Data via HTTP

GET Ships

Ships

GET http://127.0.0.1:5000/v3/ships

Params | Authorization | Headers (7) | Body | Pre-request Script | Tests | Settings

Query Params

KEY	VALUE
Key	Value

Body | Cookies | Headers (15) | Test Results

Pretty | Raw | Preview | Visualize BETA | JSON


```
[
    {
        "ship_id": "AMERICANCHAMPION",
        "ship_name": "American Champion",
        "ship_model": null,
        "ship_type": "Tug",
        "roles": [
            "Support Ship",
            "Barge Tug"
        ],
        "active": false,
        "imo": 7434016,
        "mmsi": 367020820,
        "abs": 571252,
        "class": 7604342,
        "weight_lbs": 588000,
        "weight_kg": 266712,
        "year_built": 1976,
        "home_port": "Port of Los Angeles",
        "status": "",
        "speed_kn": null,
        "course_deg": null,
        "position": {
            "latitude": null,
            "longitude": null
        },
        "successful_landings": null,
        "attempted_landings": null,
        "missions": [
            {
                "name": "COTS 1",
                "flight": 7
            },
            {
```

Space X Extension

Topology Builder

Requesting Data

Metrics

Multi Dimensional Metrics

Configuration

Requesting Data via HTTP


```
[
    {
        "ship_id": "AMERICANCHAMPION",
        "ship_name": "American Champion",
        "ship_model": null,
        "ship_type": "Tug",
        "roles": [
            "Support Ship",
            "Barge Tug"
        ],
        "active": false,
        "imo": 7434016,
        "mmsi": 367020820,
        "abs": 571252,
        "class": 7604342,
        "weight_lbs": 588000,
        "weight_kg": 266712,
        "year_built": 1976,
        "home_port": "Port of Los Angeles",
        "status": "",
        "speed_kn": null,
        "course_deg": null,
        "position": {
            "latitude": null,
            "longitude": null
        },
        "successful_landings": null,
        "attempted_landings": null,
        "missions": [
            {
                "name": "COTS 1",
                "flight": 7
            },
            {
```

Space X Extension

Topology Builder

Requesting Data

Metrics

Multi Dimensional Metrics

Configuration

Requesting Data via HTTP


```
2   {
3       "ship_id": "AMERICANCHAMPION",
4       "ship_name": "American Champion",
5       "ship_model": null,
6       "ship_type": "Tug",
```

```
14      "abs": 571252,
15      "class": 7604342,
16      "weight_lbs": 588000,
17      "weight_kg": 266712,
18      "year_built": 1976,
19      "home_port": "Port of Los Angeles",
20      "status": "",
21      "speed_kn": null,
22      "course_deg": null,
23      "position": {
24          "latitude": null,
25          "longitude": null
26      },
27      "successful_landings": null,
28      "attempted_landings": null,
29      "missions": [
30          {
31              "name": "COTS 1",
32              "flight": 7
33          },
34          {
```

Space X Extension

Topology Builder

Requesting Data

Metrics

Multi Dimensional Metrics

Configuration

Requesting Data via HTTP

```
2    {
3        "ship_id": "AMERICANCHAMPION",
4        "ship_name": "American Champion",
5        "ship_model": null,
6        "ship_type": "Tug",
```

```
19       "home_port": "Port of Los Angeles",
20       "status": "",
21       "speed_kn": null,
```

```
21       "speed_kn": null,
22       "course_deg": null,
23       "position": {
24           "latitude": null,
25           "longitude": null
26       },
27       "successful_landings": null,
28       "attempted_landings": null,
29       "missions": [
30           {
31               "name": "COTS 1",
32               "flight": 7
33           },
34           {
```

Adding additional Metrics

- Metric Definition – Minimum Requirements
  - Unique Key
  - Unit
  - Aggregation
  - Display Name

```
"metrics": [
  {
    "timeseries": {
      "key": "fuel",
      "unit": "Unspecified",
      "aggregation": "avg",
      "displayname": "Fuel"
    }
  }
```

Hands-on: Version 0.1

Adding additional Metrics

- Metric Definition – Minimum Requirements
  - Unique Key
  - Unit
  - Aggregation
  - Display Name

```
"metrics": [
  {
    "timeseries": {
      "key": "fuel",
      "unit": "Unspecified",
      "aggregation": "avg",
      "displayname": "Fuel"
    }
  }
```

Multi Dimensional Metrics

- Metric Definition – Minimum Requirements
  - Unique Key
  - Unit
  - Aggregation
  - Display Name
  - Dimensions

```
"metrics": [
  {
    "timeseries": {
      "key": "fuel",
      "unit": "Unspecified",
      "aggregation": "avg",
      "displayname": "Fuel"
    }
  }
```

Multi Dimensional Metrics

- Metric Definition – Minimum Requirements
  - Unique Key
  - Unit
  - Aggregation
  - Display Name
  - Dimensions

```
"metrics": [
  {
    "timeseries": {
      "key": "thrust",
      "unit": "Percent",
      "displayname": "Thrust",
      "dimensions": [ "engine" ]
    }
```

Space X Extension

Multi Dimensional Metrics

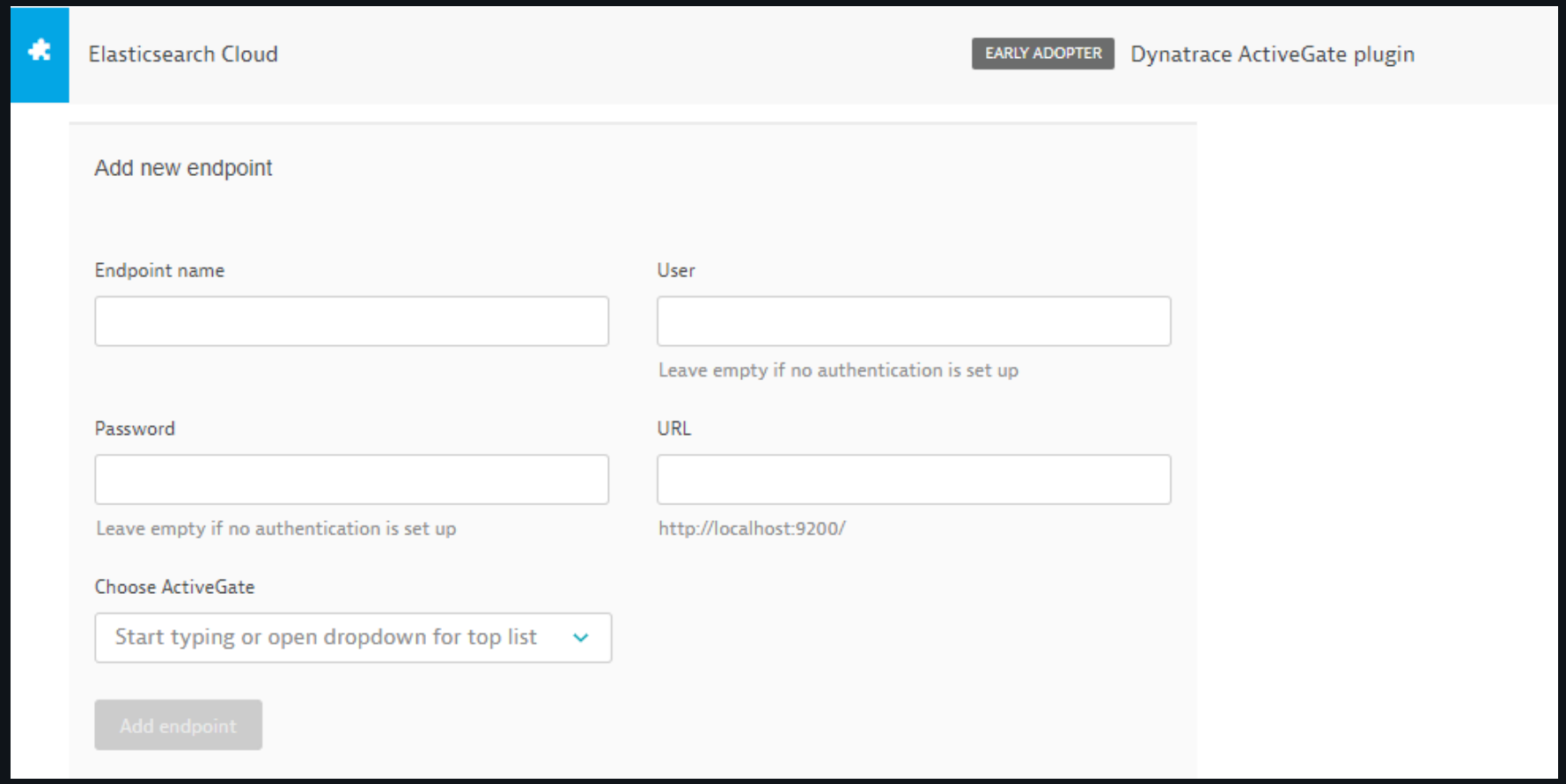
Configuration

Key Metrics

Charting

Key Charts

Make your extension configurable



Hands-on: Version 0.2

Space X Extension

Multi Dimensional Metrics

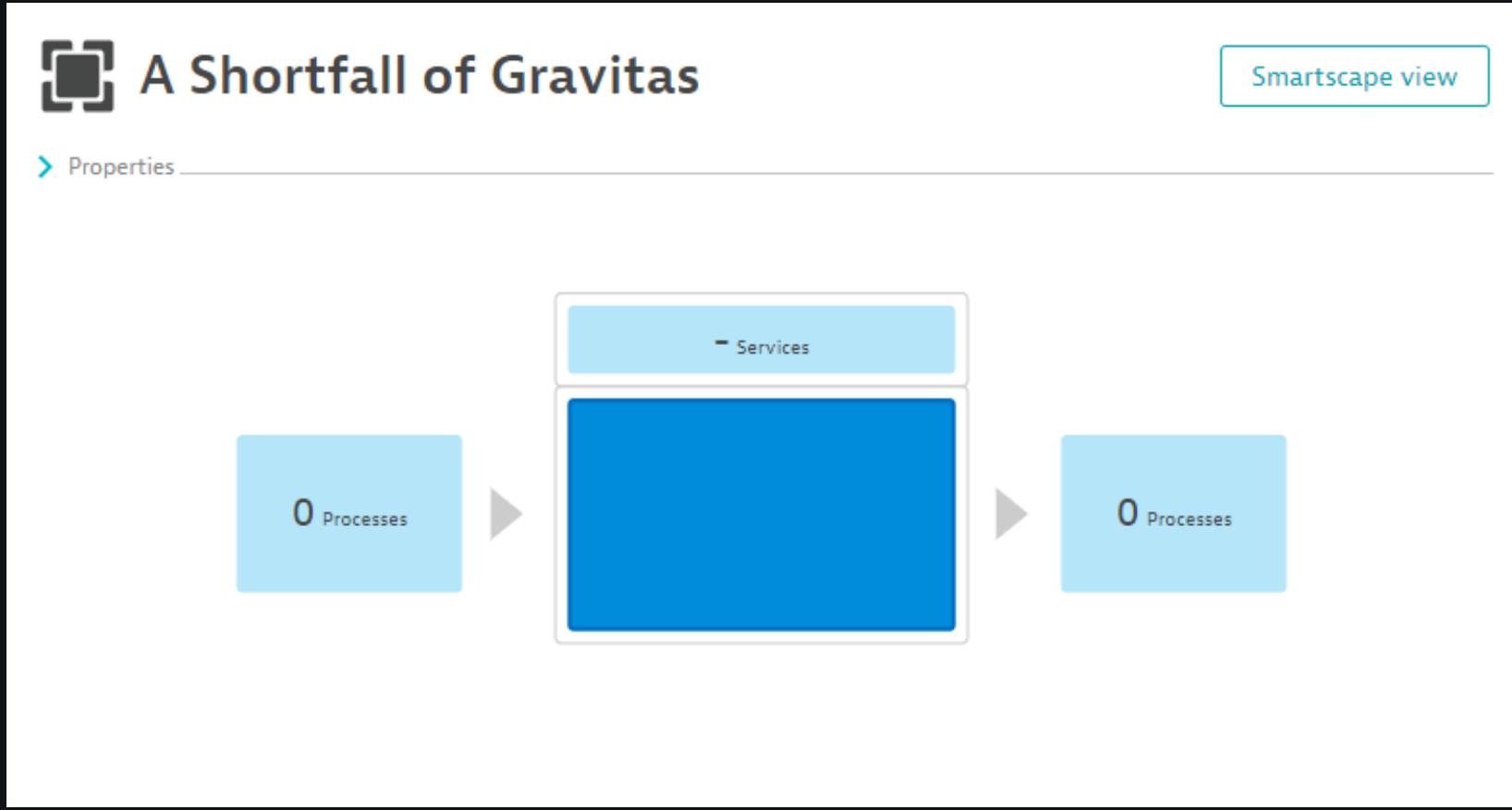
Configuration

Key Metrics

Charting

Key Charts

Key Metrics



Space X Extension

Multi Dimensional Metrics

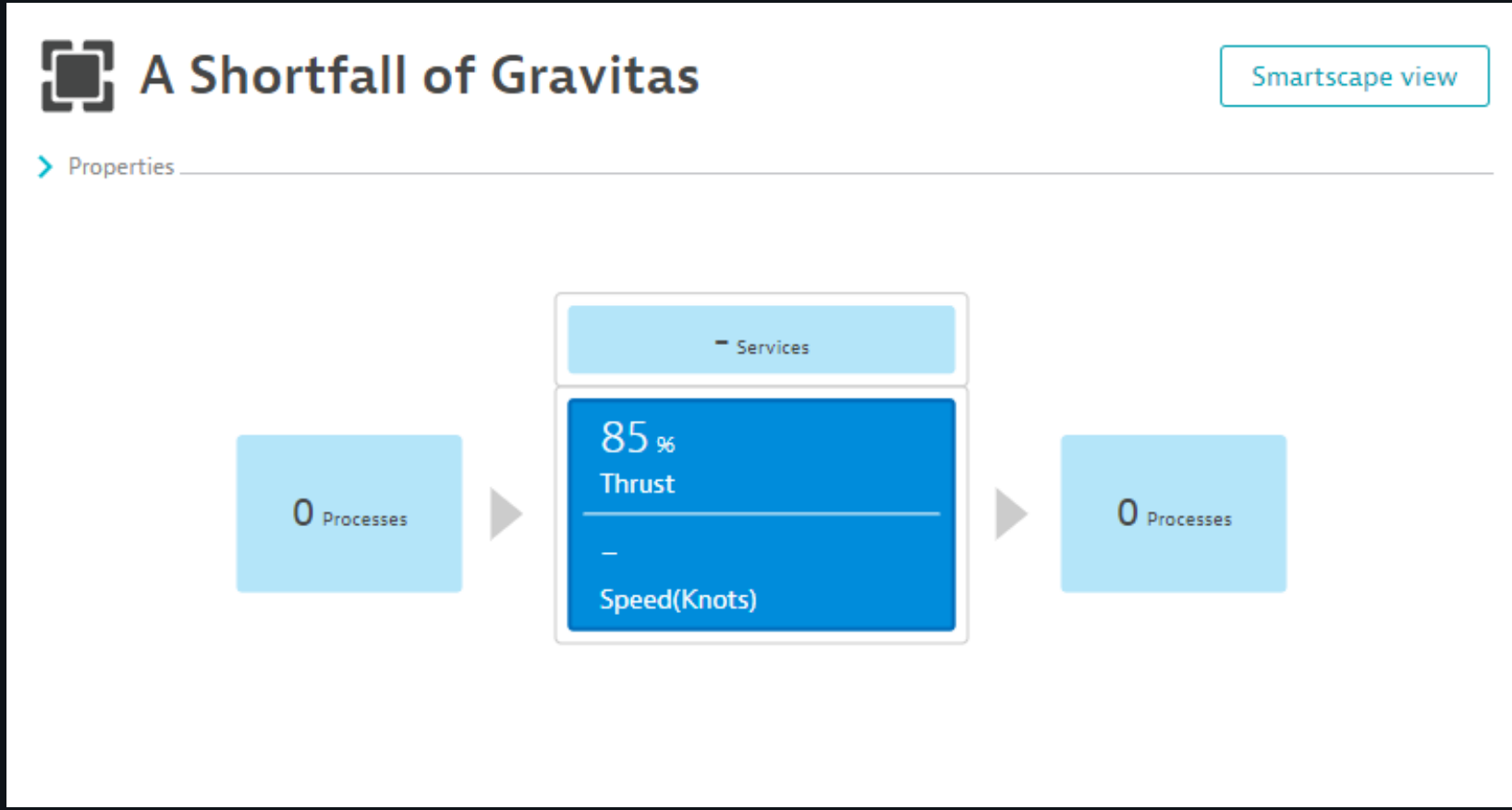
Configuration

Key Metrics

Charting

Key Charts

Key Metrics



Space X Extension

Multi Dimensional Metrics

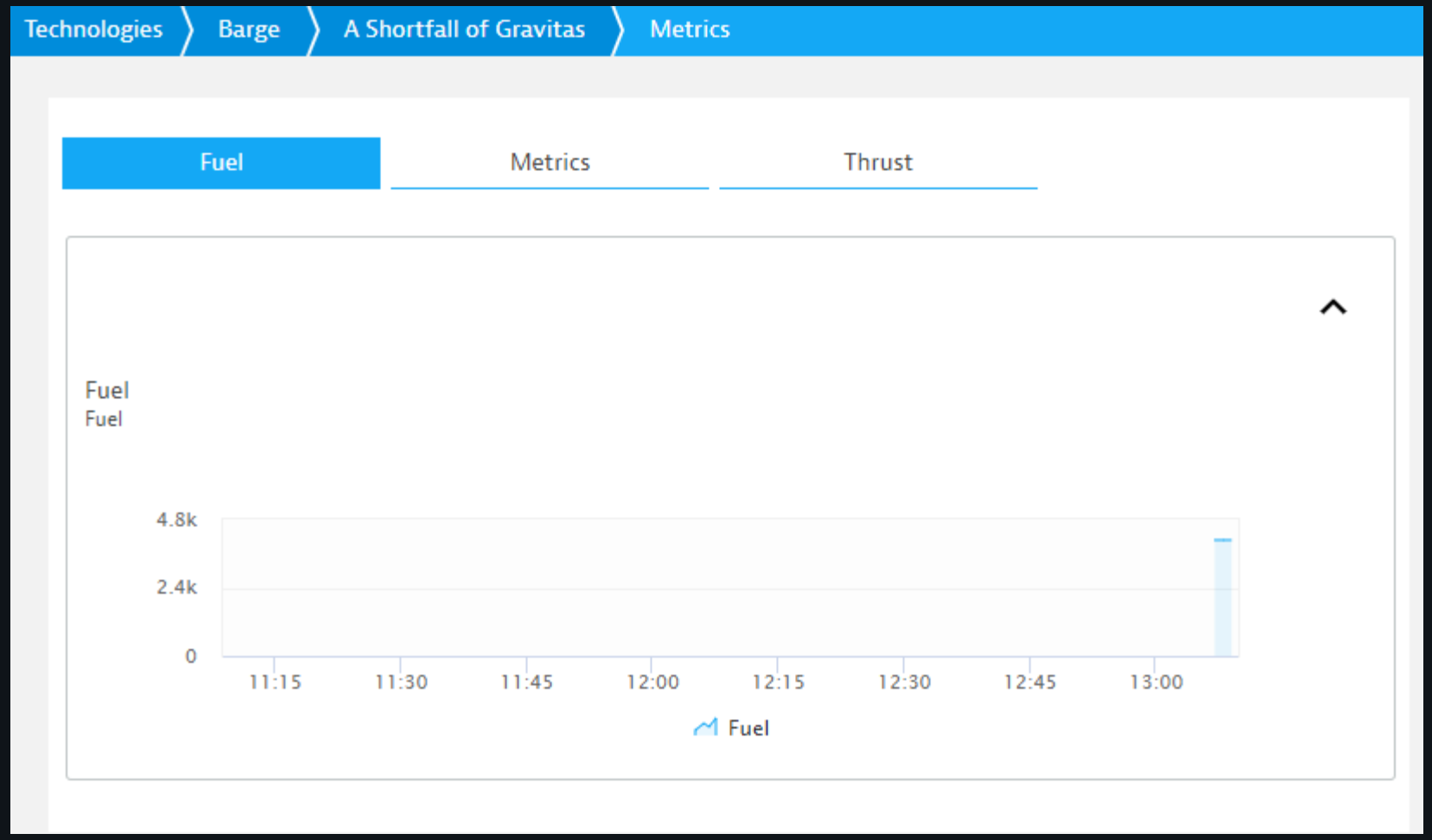
Configuration

Key Metrics

Charting

Key Charts

Charting



Space X Extension

Multi Dimensional Metrics

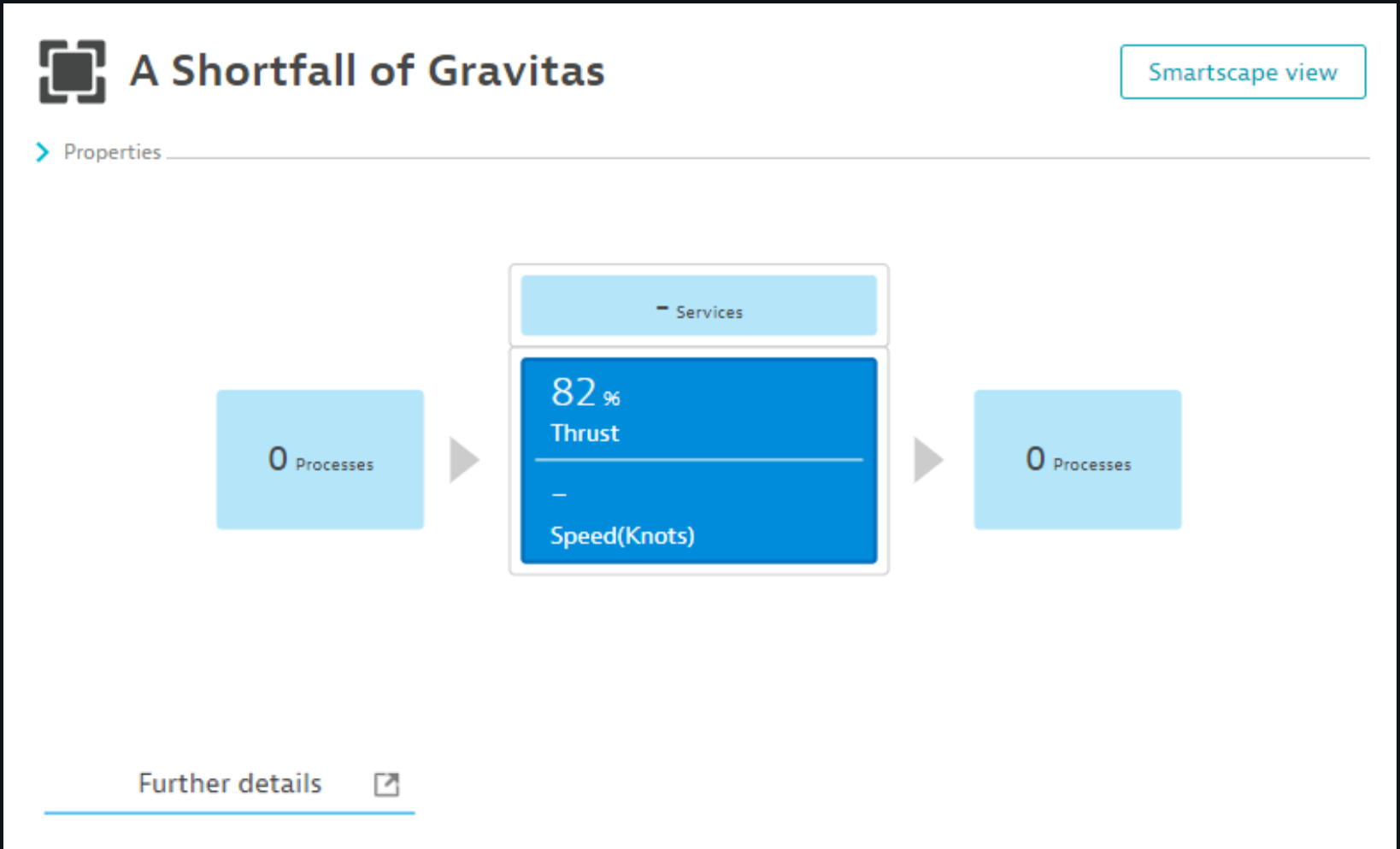
Configuration

Key Metrics

Charting

Key Charts

Key Charts



Space X Extension

Multi Dimensional Metrics

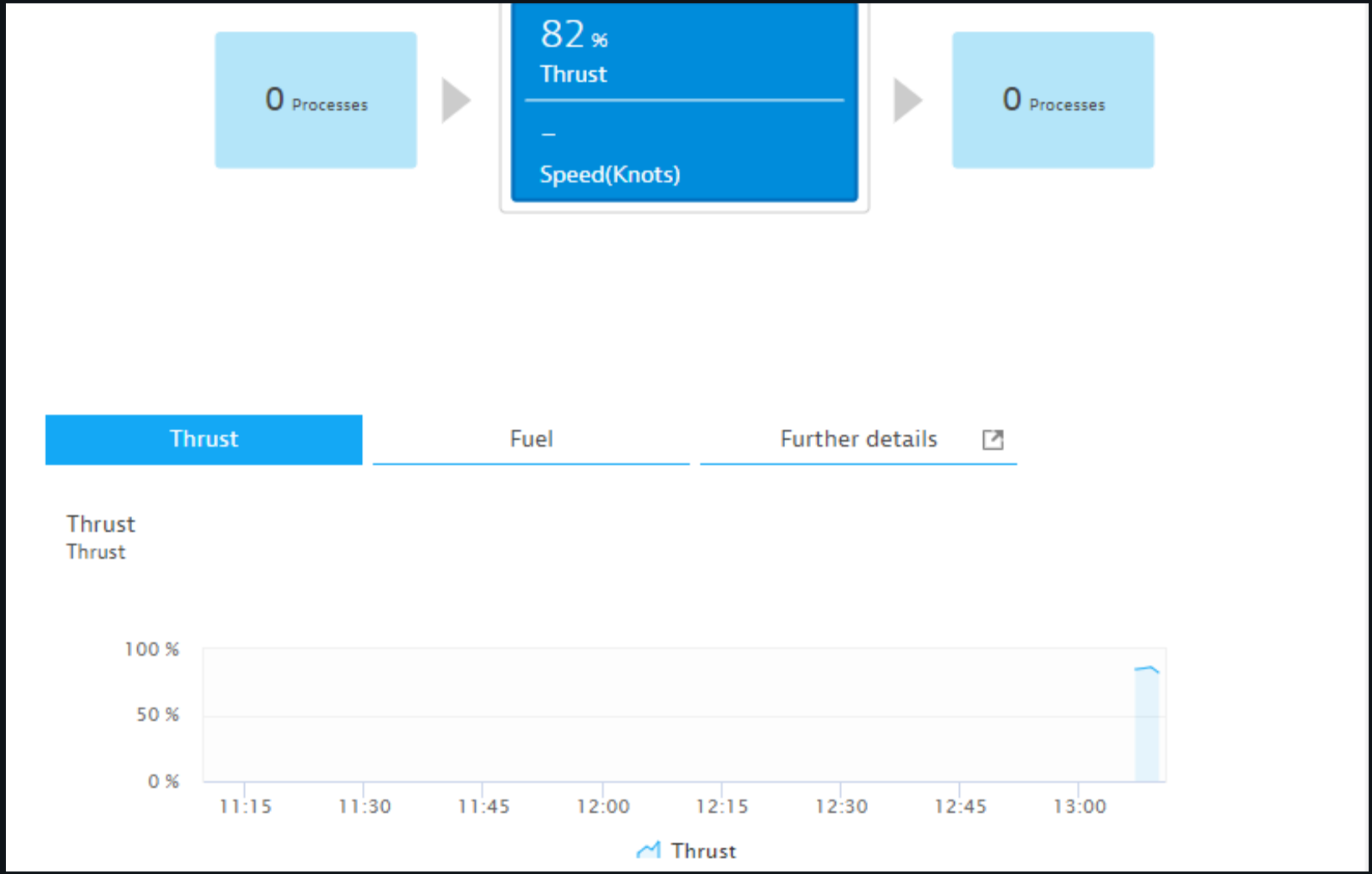
Configuration

Key Metrics

Charting

Key Charts

Key Charts



Hands-on: Version 0.3

AG Extension best practice

Dimensions

Limitations

When to use

Licensing

Cross platform

Dimensions

- Multiple dimensions
- Changing requires deletion of extension
- 100 limit (version 1.210 increases it to 1000)

```
{
    "timeseries": {
        "key": "LBVirtualServer.vsreq",
        "unit": "Count",
        "displayname": "Requests",
        "dimensions": ["VirtualServer"],
        "topxconfig": {
            "limit": 1000
        }
    }
},
```

- DDU license

Other limitations

- 100 max events per execution
- Error reporting
  - OK or Error
  - Logs
- Cannot send data to existing entities
- Only predefined metric names in the plugin.json
- Execution needs to finish within 1 minute
- 500 Custom devices per endpoint
- 100 IP/Port combinations per custom device
  - It is possible to only assign an IP

AG Extension best practice

Dimensions

Limitations

When to use

Licensing

Cross platform

When to use

- Cannot run OneAgent extensions
  - Appliances
  - Solaris
  - AIX
- Send to a custom device or as a task scheduler
- The metrics are known beforehand
  - Alternative - API
- Davis consuming the data
  - Service flow
  - Unmonitored host
  - Proxy (still to come)

AG Extension best practice

Dimensions

Limitations

When to use

Licensing

Cross platform

Licensing

- Davis Data Units (DDUs)
- 0.001 DDU per data point sent
- Metrics * dimensions * devices * interval
- Global free tier (200.000 per year)
- Free tier for OneAgent instrumented servers
- Only ACTIVELY consumed metrics are relevant
- Properties and Tags are not metrics

https://www.dynatrace.com/support/help/reference/monitoring-consumption/davis-data-units/

Cross platform

- When do you need to build it on several platforms?
- Native libraries
- Platform dependencies
  - Windows
  - Linux

R&D

Roadmap

Coming soon and later

Democratized extensibility

- Environment configuration
- Monitoring configuration

SNMP	WMI	Prometheus
AWS	Azure	GCP
Telegraf	Statsd	JMX

Available or coming soon

Extensible Davis AI & all data in scale

- Python coded extensions
- Log extensibility (detection, ingest and processing)
- More extensibility for events (SNMP Traps, Windows Events)
- Extensibility in containerized environments
- More declarative capabilities
- Containerized extensibility
- More coding capabilities

Planned for later

R&D

Roadmap

Coming soon and later

Democratized extensibility

- Environment configuration
- Monitoring configuration

SNMP	WMI	Prometheus
AWS	Azure	GCP
Telegraf	Statsd	JMX

Available or coming soon

Extensible Davis AI & all data in scale

- Python coded extensions
- Log extensibility (detection, ingest and processing)
- More extensibility for events (SNMP Traps, Windows Events)
- Extensibility in containerized environments
- More declarative capabilities
- Containerized extensibility
- More coding capabilities

Planned for later

Extension framework 2.0

Hands on 1/2

Hands on 2/2

Steps for creating an EF 2.0 extension which contains a dashboard 1/2

- Create a dashboard with one or more tiles and can call it "extension 2.0"
  - Grab the ID from the URL https://xxxxxx.sprint.dynatracelabs.com/#dashboard;gtf=-2h;gf=all;id=b13472ec-6e7c-4e8d-baa3-df40a66c7ab9
- Export the dashboard via the configuration API to dashboards/dashboard.json
  - Go to the dashboards section, expand /dashboards/{ID} (3rd link) and press "try it out.
  - Enter your dashboard ID.
- Change the name of the dashboard in the file to "extension 2.0.1"
- Remove the "ID" : "xxxx" line from the file
- At the root of your project create an extension.yaml file

```
name: com.dynatrace.demo.extension
version: 1.0.0
minDynatraceVersion: "1.209"
author:
    name: demo
dashboards:
 - path: "dashboards/dashboard.json"
```

- Zip the yaml file and the dashboards folder under extension.zip

Steps for creating an EF 2.0 extension which contains a dashboard 2/2

- Sign the extension with openssl via command prompt
  - Cd to your project directory
  - openssl genrsa -out customer_private.pem 4096
  - openssl dgst -sha256 -sign customer_private.pem -out extension.zip.sig extension.zip
- Zip extension.zip and extension.zip.sig under demoextension.zip
- Upload the extension via the API
  - Use environment API 2.0, extensions 2.0, POST extension (2nd link) and try it out
  - Choose your demoextension.zip file
- Enable the extension via the API
  - use the 8th link: POST extensions/extensionname/environmentConfiguration.
  - Enter the name of the extension com.dynatrace.demo.extension and enter the correct version number (1.0.0).

Thank you!