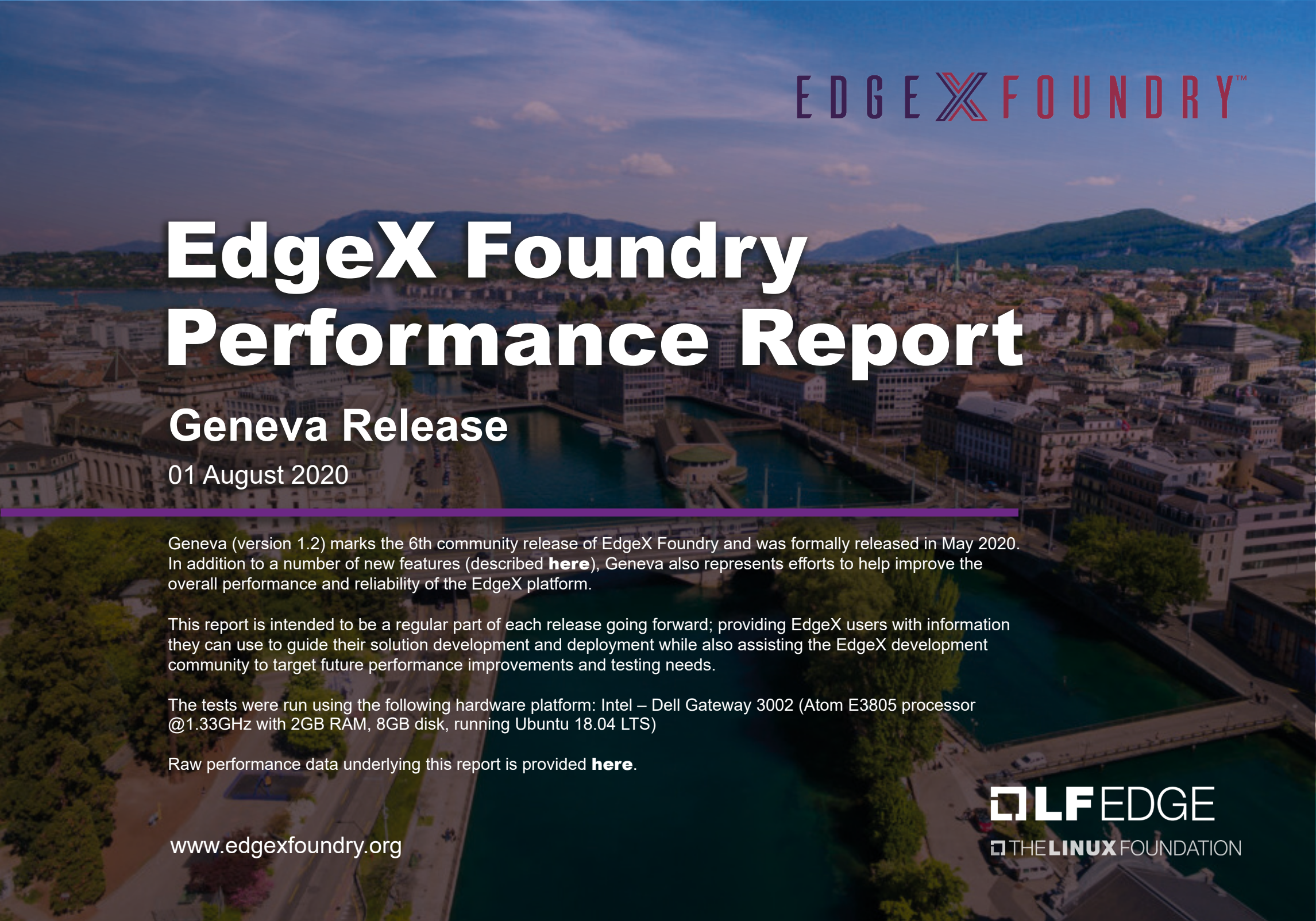


EDGE X FOUNDRY™

EdgeX Foundry Performance Report

Geneva Release

01 August 2020

Geneva (version 1.2) marks the 6th community release of EdgeX Foundry and was formally released in May 2020. In addition to a number of new features (described **here**), Geneva also represents efforts to help improve the overall performance and reliability of the EdgeX platform.

This report is intended to be a regular part of each release going forward; providing EdgeX users with information they can use to guide their solution development and deployment while also assisting the EdgeX development community to target future performance improvements and testing needs.

The tests were run using the following hardware platform: Intel – Dell Gateway 3002 (Atom E3805 processor @1.33GHz with 2GB RAM, 8GB disk, running Ubuntu 18.04 LTS)

Raw performance data underlying this report is provided **here**.

www.edgexfoundry.org

LF EDGE

THE LINUX FOUNDATION

Deployment Options

The EdgeX Foundry platform is comprised of a collection of microservices. While a number of services are used in almost all deployments of the platform, the specific collection of services used are dependent on the use case, solution architecture, and user discretion.

EdgeX provides flexibility in its architecture that offers many options for deployment and use. The EdgeX microservices are designed to be location independent so can be distributed between computing nodes where necessary, however the EdgeX Core services are usually expected to be ran on all nodes. In very austere environments, a deployment many only include a device service and a few core services.

Therefore, this report provides resource usage based on the “general” deployment and “minimal deployment”. The general deployment includes all those services used together as envisioned by the original architects of the platform and inclusive of one example device service.

Specifically, the general deployment includes the following services and infrastructure:

Service	EdgeX Service Name	Required or Optional
Consul	edgex-core-consul	Optional – configuration can be obtained from local file
Database	edgex-redis	A database is required and Redis is the default implementation
Core Data	edgex-core-data	Required
Core Metadata	edgex-core-metadata	Required
Core Command	edgex-core-command	Optional – only required in actuation use cases
App Service Configurable	edgex-app-service-configurable-rules	Optional – for pipeline data processing and exporting to application endpoints
Support Notifications	edgex-support-notifications	Optional – services do not have to notify via central service
Support Scheduler	edgex-support-scheduler	Optional – services do not have to use a central scheduler (Device Services embed their own)
Device Virtual	edgex-device-virtual	Optional – any number of device services can be connected. Device Virtual serves as the representative example

The minimal deployment includes a subset of services that support a basic deployment:

Service	EdgeX Service Name	EdgeX Required or Optional
Database	edgex-redis	A database is required and Redis is the default implementation
Core Data	edgex-core-data	Required
Core Metadata	edgex-core-metadata	Required
Device Virtual	edgex-device-virtual	A deployment would include at least one device service and the virtual device service serves as the representative example

Service Executable and Image Size (Footprint)

The EdgeX microservices are typically implemented in Go or C and then compiled into an executable which has a size, otherwise known as footprint, as it sits on disk.

Additionally, each microservice executable, along with a base operating system, any required configuration and supporting infrastructure, can be “containerized” for easier deployment and orchestration with the other EdgeX microservices. As an example, the EdgeX microservices are built into Docker images for deploying as Docker containers. Each Docker image also has a footprint size as it sits on disk. This is typically much larger as it incorporates a base OS, infrastructure, configuration, etc, but as described above can greatly simplify deployment because it adds platform independence.

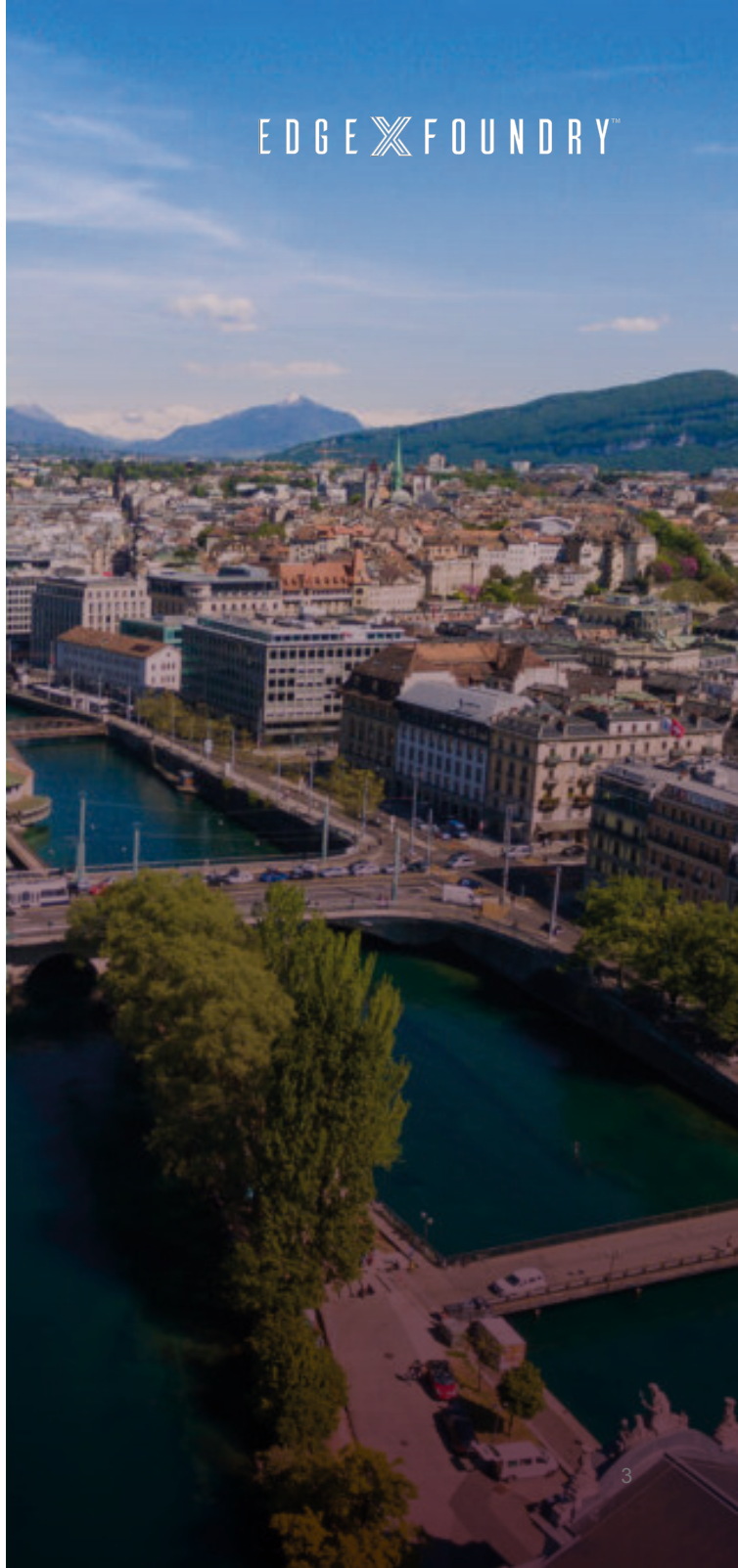
Depending on the use case, a user of EdgeX may choose to deploy EdgeX in either containerized or non-containerized form. The raw executables could also be used to deploy and orchestrate EdgeX using an alternative mechanism. Both the non-containerized “Executable” and containerized "Image” footprint data is reported here.

General Deployment Footprint

Footprint in MB	Image	
Microservice	Footprint	Executable
edgex-core-consul	139.26	N/A
edgex-core-data	23.80	15.65
edgex-core-metadata	14.42	14.41
edgex-core-command	12.99	12.98
edgex-support-notifications	14.45	13.12
edgex-support-scheduler	13.12	13.12
edgex-app-service-configurable-rules	29.84	21.18
edgex-device-virtual	20.84	15.23
edgex-redis	29.78	N/A
TOTAL	**298.0**	

Minimal Deployment Footprint

Footprint in MB	Image	
Micro service	Footprint	Executable
edgex-redis	29.78	N/A
edgex-core-data	23.80	15.65
edgex-core-metadata	14.42	14.41
edgex-device-virtual	20.84	15.23
TOTAL	**89.0**	



CPU Usage

The CPU usage of each container was measured on startup of the service. It is measured as a percentage of CPU available as reported by the Docker Engine. Because the CPU characteristics of the different chip architectures vary, the percentage of CPU usage can differ widely. CPU usage on the Intel Atom Processor (E3805 1.33GHz 1MB L2 cache) is around 15% - inclusive of infrastructure elements (database, configuration/registry, etc.).

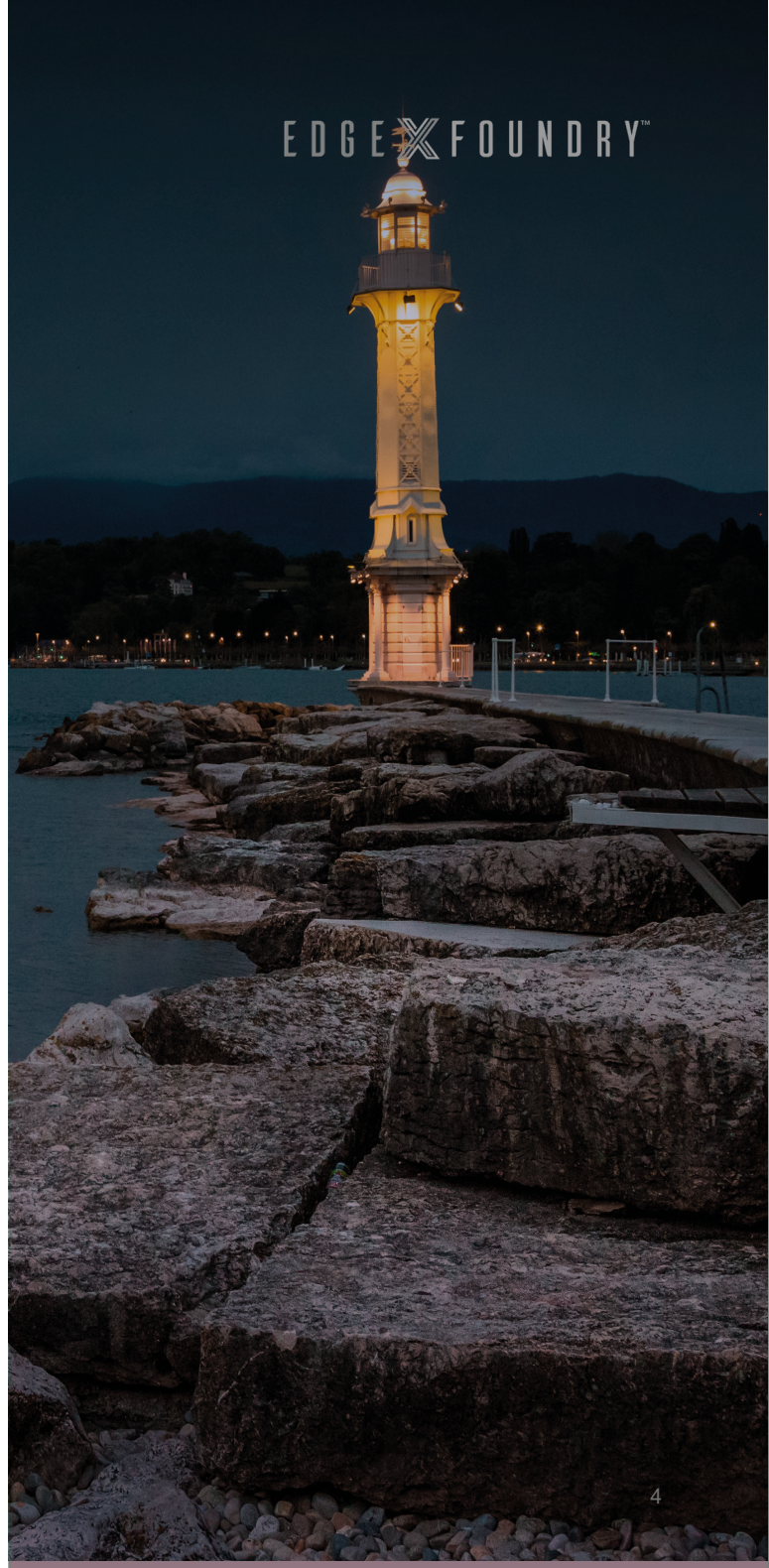
General Deployment CPU Usage

Microservice	CPU usage at startup (%)
edgex-core-consul	2.29%
edgex-core-data	1.80%
edgex-core-metadata	2.01%
edgex-core-command	1.74%
edgex-support-notifications	1.70%
edgex-support-scheduler	1.52%
edgex-app-service-configurable-rules	0.15%
edgex-device-virtual	3.60%
edgex-redis	0.76%
TOTAL	**15.5%**

Minimal Deployment CPU Usage

Microservice	CPU usage at startup (%)
edgex-redis	0.47
edgex-core-data	1.98
edgex-core-metadata	1.87
edgex-device-virtual	0
TOTAL	**4.32**

***Future Consideration**: In general, the services consume a lot of CPU as they startup and so the measure of usage at startup can be a good upper bound for many services. However, future performance tests will also test the CPU usage periodically throughout a period of time and at peak sensor data ingestion and device actuation times.*



Memory Usage

Again, the memory usage of each container was measured on startup of the service. It is measured in MB used as reported by the Docker Engine. **Memory usage**, in a containerized environment, **is around 56MB on the Intel platform** (inclusive of infrastructure elements).

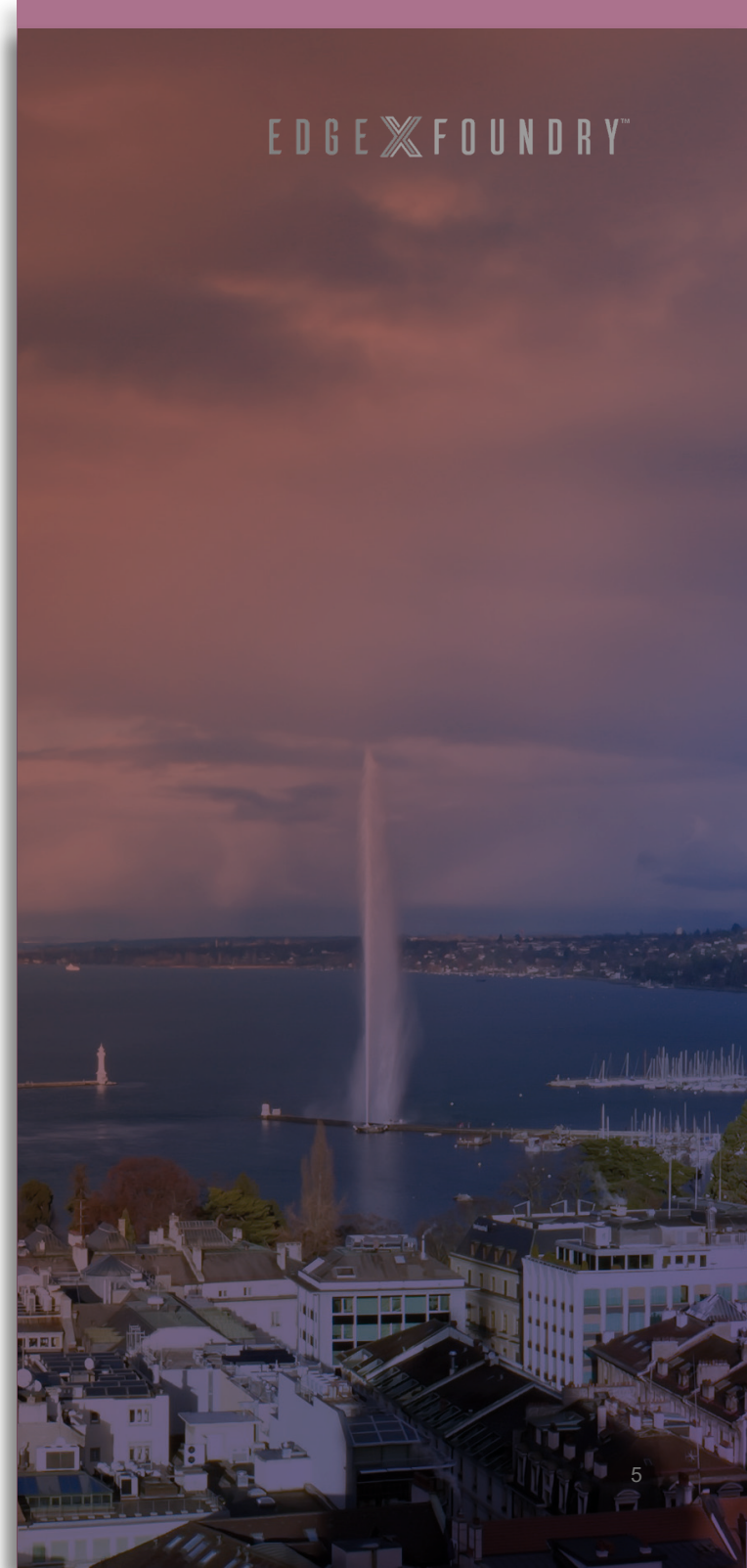
General Deployment Memory Usage

Microservice	Memory usage at startup (MB)
edgex-core-consul	17.75
edgex-core-data	5.11
edgex-core-metadata	5.94
edgex-core-command	3.67
edgex-support-notifications	4.30
edgex-support-scheduler	4.39
edgex-app-service-configurable-rules	5.98
edgex-device-virtual	6.49
edgex-redis	2.14
TOTAL	**56.0**

Minimal Deployment Memory Usage

Microservice	Memory usage at startup (MB)
edgex-redis	2.14
edgex-core-data	5.11
edgex-core-metadata	5.94
edgex-device-virtual	6.49
TOTAL	**31.0**

***Future Consideration**: In general, the services consume various amounts of memory during operation. Future performance tests will also explore the memory usage at peak usage times such as during sensor data ingestion and device actuation.*



Operational Latency

Startup and Ping Operations

Currently, the startup (a.k.a. bootstrap) time and response to the HTTP ping request are the only two operational performance measures collected across all services. Additionally, the time to export the data to an endpoint is also measured.

The time to pull the Docker container image, create the container and start all of the EdgeX containers is significantly higher – especially on more resource constrained platforms. The time to pull, create and start EdgeX containers on the 2GB (RAM) Intel platform is just under a minute.

Microservice	Startup Time (Binary)	Startup Time (Container+Binary)	Ping (ms)
edgex-core-data	1.31185939 s	35.652928829193115 s	14.631
edgex-core-metadata	1.431470976 s	30.986926794052124 s	8.435
edgex-core-command	787.06282 ms	36.832406759262085 s	11.033
edgex-support-notifications	4.281586174 s	29.65292477607727 s	9.812
edgex-support-scheduler	6.117744992 s	30.341880798339844 s	9.155
edgex-export-virtual	3.607480699 s	44.42702794075012 s	11.271
TOTAL		**44.42702794075012 s**	

Data Export

Recording of the time it takes to send EdgeX data from the “south side” (sensors) through the platform to the “north side” (applications). The measurement includes the time it takes the virtual device service to create the Event/Reading, send it to (and through) Core Data, and have the Application Service read and prepare the data for a northbound system. The event is extracted from the Core Data supplied event message topic, sent to its designated endpoint, and marked as “pushed” in Core Data.

The total average data export time for this scenario is measured as 35ms.

Methodology

The following notes describe how the performance data described in this report was captured. The services were run with their default configurations and the Virtual Device was allowed to run and produce sensor data in its default manner (generating data every 30 seconds for a variety of mimicked sensors).

The CPU and memory usage data was captured from Docker Engine via docker stats after successful start-up of EdgeX using the release Docker Compose files. Data export testing was run 15 times – that is generating 15 event/readings through Core Data to the Application Services layer via the Virtual Device Service.

EDGE X FOUNDRY™

EdgeX Foundry is an open source, vendor neutral, flexible, interoperable, software platform at the edge of the network, that interacts with the physical world of devices, sensors, actuators, and other IoT objects. In simple terms, EdgeX is edge middleware - serving between physical sensing and actuating "things" and our information technology (IT) systems.

The EdgeX platform enables and encourages the rapidly growing community of IoT solution providers to work together in an ecosystem of interoperable components to reduce uncertainty, accelerate time to market, and facilitate scale.

Geneva Release

The EdgeX v1.x series of release include important new features as well as improvements and hardening of the existing EdgeX functionality. Highlights of the Geneva release include:

- Dynamic or automatic device on-boarding – for protocols where it makes sense, this allows EdgeX to automatically provision new sensors and have the sensor data start to flow through EdgeX
- **Alternate messaging support – where applicable, allows adopters to use any messaging implementation under the covers** of EdgeX to include MQTT, 0MQ, or Redis Streams
- Better type information is associated to sensor data – allowing analytics packages and other users of EdgeX sensor data to more easily digest the data and aid in transformations on the data
- REST device service
- Batch and send export – allowing sensor data to be sent to cloud, on-prem or enterprise systems at designated times
- Support for MQTTS and HTTPS export
- Redis as the default DB – deprecating MongoDB for license, footprint/performance, and ARM support reasons
- Adding the Kuiper rules engine – a new rules engine that is smaller and faster and written in Go which replaces EdgeX's last Java micro service
- Improved Security
- Interoperability testing
- Improved DevOps CI/CD – now using Jenkins Pipelines to produce EdgeX Foundry project artifacts

Have you got any questions?

This document was produced by IOTech Systems – the Edge Software Company whose products include Edge Xpert which is an enhanced, productized and commercially-supported implementation of EdgeX Foundry.

If you have questions about this report or if you need more information on IOTech or Edge Xpert, email **info@iotechsys.com** or visit the website **www.iotechsys.com**

Contact Us

For general information about Edge X, EdgeX Foundry, or membership inquiries, please email **info@lfedge.org**

Visit our website at

www.edgexfoundry.org

LF EDGE

THE LINUX FOUNDATION