

Realtime Workload Metrics - Feature Design Candidate

Status: In Progress

Sponsor User: NS1

Date of Submission: 17 Jul 2023

Submitted by: Joe Pearson

Affiliation(s): IBM

<Please fill out the above fields, and the Overview, Design and User Experience sections below for an initial review of the proposed feature.>

Scope and Signoff: (to be filled out by Chair)

<Please fill out the Overview, Design and User Experience sections for an initial review of the proposed feature.>

Overview

There are limited built-in metrics available about an edge node, its host, and workloads. Traditional approaches include streaming logs to a remote host, or surfacing them so they can be queried on the edge node by a remote operator logging in. Each of those approaches has drawbacks and are not edge native.

Design

If an application were to be installed on an edge node, preferably delivered by Open Horizon, that could query the system for information, surface it, and make the data available in an efficient and edge-native manner, that would be ideal. This may mean updating the node properties, and it may mean making the information remotely queryable without the operator logging in to the edge node.

The design includes 3 layers described below:

- The platform functionality
- The node functionality
- The query and monitoring functionality

This Feature Design candidate will deliver a [functioning end-to-end example and documentation](#) demonstrating how to deliver and configure an open-source version of AnyLog. This code will deliver and connect a data collection and querying network consisting of a master node, a query node, and two or more operator nodes. Future iterations may include a version using non-containerized node agents, and a script that installs and integrates the open-source version of AnyLog beside an [All-in-One](#) Open Horizon deployment instance in similar fashion to how FDO is integrated.

The Platform Functionality - Extending Open Horizon as a Platform:

AnyLog extends the Open Horizon functionality delivered to the edge as a platform:

- A shared metadata layer (hosted on blockchain or a master node) that contain policies shared among participating nodes. For example:
 - Policies representing the members of the network.
 - Policies representing the schemas used.
 - Policies representing configurations.
 - Policies representing nodes and users permissions.
 - Any metadata that needs to be shared among nodes of the network.
- A Peer to Peer and secure network using the AnyLog protocol allowing nodes to exchange messages.

The Node Functionality - Extending the functionalities of nodes deployed by Open Horizon:

AnyLog extends the Open Horizon functionality delivered to the individual nodes by using the platform functionality such that:

- Data that needs to be monitored will be persistent in a local database - nodes collect and monitor the target metrics.
- The schemas that are used to store the data are shared among all participating nodes.
- Each node is extended to include a rule engine that can act on data and status events.
 - Using the rule engine - thresholds are monitored to trigger alerts when needed.
 - Using the rule engine - old data is removed and archived to avoid storage overload.
- Each node is extended to include southbound connectors (to ingest data) and northbound connectors (to share data).

KubeArmor running on the edge node provides visibility and protection for all the processes, files, or network operations in the containers as well as those running directly on the host. See [KubeArmor integration repo](#). In this feature, KubeArmor (when present) can transmit (define how) collected metrics to the AnyLog code running on the Node.

The query and monitoring functionality

Nodes members of the network, as well as applications connected to nodes in the network, are able to view all the monitored data as if it is a single and unified collection of data.

Practically, nodes view a virtual database based on the schema published by the shared metadata layer and can issue queries to the data as if the data is centralized.

The query or monitoring can view an entire network as a single machine, or dynamically partition the network to satisfy the user view by criteria's determined by the users (and represented in the shared metadata policies).

For example: by locations, by type of software deployed, by owners etc.

[Joe Pearson](#) Can we label arbitrary groups or data points by purpose: APM, Security, etc.

NS1 will provide an API endpoint and help define when how, and what information will be transmitted from the Nodes over AnyLog into NS1 for Node and network visibility and analytics.

User Experience

<Describe which user roles are related to the problem AND the solution, e.g. admin, deployer, node owner, etc. If you need to define a new role in your design, make that very clear. Remember this is about what a user is thinking when interacting with the system before and after this design change. This section is not about a UI, it's more abstract than that. This section should explain all the aspects of the proposed feature that will surface to users.>

User experience is similar to the experience with a cloud/centralized solution:

- From a single point, the distributed data can be queried as if the data is hosted in a centralized database.
 - A user selects a database from a list of virtual databases.
 - A user selects a table from a list of virtual tables.
 - A user issues a query to the table.
 - Optional - The default behaviors is a reply from all nodes with relevant data, However, a user can specify a subset of nodes (for example: nodes deployed in a region or nodes with a named data owner).
- From a single point, all the resources are monitored and managed as if the resources are hosted in a single machine.
 - User can issue a status request from all nodes or to a subset of nodes (for example: nodes deployed in a region or nodes with a named data owner).
 - Users can identify a node to host pushed data (from the edge nodes) representing current status (an equivalent to a repeatable query).
- Using the rule engine, users and processes can be alerted by events on the individual nodes or on the aggregator node.

Command Line Interface

<Describe any changes to the hzn CLI, including before and after command examples for clarity. Include which users will use the changed CLI. This section should flow very naturally from the User Experience section.>

Are there any ways to optionally extend the CLI when components are installed? If not, they we should avoid this.

The lower level AnyLog functionality is enabled by a CLI, this can extend the hzn CLI.

AnyLog CLI includes dynamic help with links to help pages on GitHub - all of that can be available as an extension of the hzn CLI.

Additional information:

- Nodes in the AnyLog network are configured such that commands and queries can be provided using REST. Therefore it is simple to integrate to existing and new applications without dependencies on existing infrastructure or setups.
- Because of the decentralization nature of the AnyLog Network - any node or application can act as a point of access to the entire data set and the monitored status of all the member nodes.
- AnyLog provides a web GUI that is optimized to the AnyLog API calls and data queries. It only requires a browser, can be installed on any node and can serve as a monitoring tool for network managers and as a training tool for administrators and developers showing how to interact with nodes in the network.

External Components

<Describe any new or changed interactions with components that are not the agent or the management hub.>

Installing the Anylog agent on an edge node should provide metrics collection and surfacing.

This can be done by a policy representing the metrics and associating the node with a metrics policy.

The metrics policy can be identical on all nodes or specific to a node or a group of nodes.

Affected Components

<List all of the internal components (agent, MMS, Exchange, etc) which need to be updated to support the proposed feature. Include a link to the github epic for this feature (and the epic should contain the github issues for each component).>

N/A

Security

<Describe any related security aspects of the solution. Think about security of components interacting with each other, users interacting with the system, components interacting with external systems, permissions of users or components>

Ideally, the Anylog component should not need root-level access. – That is correct.

The Anylog component maintains its own P2P network. – That is correct.

An AnyLog node can be deployed with and without security layers. If enabled - the AnyLog protocol is using keys and the blockchain to authenticate users and their permissions. The network can issue certificates to 3rd parties applications that authenticate the apps and users and determine their permissions.

APIs

<Describe and new/changed/deprecated APIs, including before and after snippets for clarity. Include which components or users will use the APIs.>

Link to Anylog docs.

- Each AnyLog instance includes a CLI option.
- Data monitored can be generated by AnyLog existing functionalities. For example, disk space, memory usage, networking status, cpu state, processes running etc. are build-in functionalities that can be leverage on each node. Additional details are in the [Monitor Nodes](#) document.
- Southbound Connectors are detailed in the [Adding Data](#) document (including services to present a node as a broker for pub-sub of a data, to subscribe to a third party broker, to receive data via REST).
- Northbound connectors are based on SQL and AnyLog CLI commands that are transferred to the network using REST.

AnyLog documentation:

- [Table of Content](#)
- [Getting Started](#)

Build, Install, Packaging

<Describe any changes to the way any component of the system is built (e.g. agent packages, containers, etc), installed (operators, manual install, batch install, SDO), configured, and deployed (consider the hub and edge nodes).>

Will be done using Open Horizon (we had a prototype Open Horizon + AnyLog working).

A detailed Docker based deployment training is available with [this link](#).

Documentation Notes

<Describe the aspects of documentation that will be new/changed/updated. Be sure to indicate if this is new or changed doc, the impacted artifacts (e.g. technical doc, website, etc) and links to the related doc issue(s) in github.>

- Document deployment with Open Horizon.
- AnyLog CLI extending the Open Horizon CLI.

[Rahul Jadhav](#) : Can we please add the documentation for all the possible ways in which the data can be ingested in to AnyLog? CC: [Moshe Shadmon](#)

Test

<Summarize new automated tests that need to be added in support of this feature, and describe any special test requirements that you can foresee.>

Simulate edge nodes deployed with AnyLog.