System Management WG Meeting: 8/6/19

Attendees: . Attendees that may have joined after the start of the meeting may not have been captured and listed.

Discussion and action items as a result of meeting in RED

Old Business

- Open Horizons sub-project
  - State of current POC
- Fuji Work
  - Metric collection by Executor – demo/discussion by Akram
    - Issue from 7/30: What about RAML – how will choice of metric be documented?
  - Set Configuration – current status
  - Still to be done
    - Start/stop/restart all done by the executor to include stop/restart of SMA
      - Executor tracks completion and returns results
    - Safe shutdown (per last meeting)
      - Let the service decide what is part of "shutdown" operation. Each service should provide a "shutdown" function – that does the exit, but service can override and have a way to prevent immediate shutdown or wait for something to finish before it shutdown. Need to work out the details of what is returned from this method.
      - From 7/30: May have it already through Destructor operation in each service. Kill it in the right manner to make sure Destructor function fires
- Management tools matrix
  - Requirements
  - Management and other tools to explore
  - Now online at https://wiki.edgexfoundry.org/download/attachments/329501/feature_matrix.xlsx?version=1&modificationDate=1565050816465&api=v2

New Business

- Edgex CLI – will be covered next week with Alex returns from vacation
  - Code currently in holding (see TSC July 10 meeting recording for demo)
  - Need review and approval by this group to TSC to move forward
  - What's MVP to accept this and bring it to edgexfoundry GitHub organization
    - Unit tests
    - Should work with EdgeX security proxy/API gateway
    - Services are currently determined by hardcoded ports
      - Need services and ports to be configurable (manifest?)
    - Other commands needed for Fuji?
  - What would be "released"? Would there be artifacts? If so, how are they to be managed, versioned, etc.?

Malini/Alex response to some questions

If you can please give some thought to and be prepared to give your impression on the following questions for the meeting:

What changes are needed to the existing tool to meet MVP and acceptance criterial for the community?

We think minimally the CLI tool should work with the Edgex security proxy.

When we first started on this effort, we were accessing the individual service endpoints.

Initial list from TSC meeting:

Some unit tests to some level of coverage

The CLI essentially forwards to the various service endpoints. So the unit tests of the various service endpoints should cover each.

The tests that would be intrinsic to the CLI itself would be

Being able to reach the EdgeX security proxy reading from the configuration parameters.

Mocking the security token

Are we missing something?

Need the available services and their ports to be configurable (by some sort of manifest)

The only thing that should be configurable should be the EdgeX security proxy.

All the remaining services should be accessible transparently via consul.

This becomes a non-issue

Windows support?

Non-issue. **Just uses Go and its dependencies.**

(Alex – would the tool check if the user has GO and if so version?

Python allowed clean little virtual environments

What is the best practice for such stuff?)

Any other commands that are MVP that are not already provided?

We do need community input here, to the best of our knowledge we have most things covered.

Anything from the Alex's roadmap that needs to be pulled in for Fuji release?

Documentation status – good enough for initial release?

We have covered in the docs tool set-up and the individual commands. The documentation will point for each sub-family of commands to the

EdgeX service API to indicate the expected output. The CLI will not massage output to allow the developer/user to see exactly what the service API command will return.

A request we had was the ability to see the curl input – which we could "echo" back to the user if verbose flag was provided.

What would the release for CLI look like? Would any thing be released or is this a do-it yourself tool for the developer?

I think it should follow the EdgeX versions and release. Owned by EdgeX, while possibly used chiefly by developers, and perhaps some by operators.

How is it versioned? Managed?

Version would be major version of EdgeX with minor version number changing if we had to patch the CLI.

But where things get interesting ..

say we have CLI from Edinburgh and we via the security API configuration point it to an instance Delhi EdgeX … we could attempt to service requests but

in case of error pass the error as is to the user with an additional statement stating that the CLI version and edgeX instance version are not in sync.

Possibly even point the user to the github link to the EdgeX CLI of the correct version.

What WG would own it?

Tools such as UI and CLI work across all the EdgeX sub-projects. Should they all be under SMA?

Alex had some ideas around meta-CLI commands .. like pointing to another database.
Pointing to another authentication and authorization service …
We did not get into the details. Something to poll the community about too.