System Management WG Meeting: 2/19/19

Attendees: Jim, Akram, AD, Trevor (Dell), Tony and Ian (Canonical), Lenny and Emad (Intel), Rodney (Beechwoods). 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

- Start/stop/restart EdgeX micro services – redesign API
  - Jim to update design doc to reflect changes below.
  - Revisit of executor location
    - Per prior meetings, we decided to have a separate repo for executors (not always go code and doesn't scale if we have to add them all to edgex-go – a la database issue)
    - Ian – agree that not all executors should go in edgex-go
      - Just put the ref impl (Docker) in the edgex-go repos
      - This is where we build Go artifacts and it is for simplicity sake
      - Simplifies build of docker container for SMA
      - Ex: K8s would not be there in edgex-go
      - All agreed with this move.
    - Review executor as separate PR put in for changes for edgex-go
      - Executor will go in the cmd – for Go/Docker/Linux ref impl
      - Executor for snaps will go in the existing snap folder - snap/local/runtime/bin- which is scripts based
      - All agreed with this plan.
  - Visit PR 926 (merged) Issues (allow it to stay merged)
    - Exceptions and issues were with code in EdgeX-go that was building executor that has not been reviewed – these are taken care of with resolution on executors above.
    - Other concerns are minor can be addressed with issues added after today's PR.
- Current on-going work
  - Add CPU usage metrics (in addition to memory usage metrics)
    - Additional API (and pass through to the services) to be added to the SMA
    - Look at abstraction around metric APIs to allow for other implementations going forward (may be walk stage after crawl with next release)
    - Example: allow memory and CPU metrics to be provided by external executable
    - To review design/implementation at next meeting
  - Add system management API to Go DS SDK
    - May need to add config key with DS to address "service name"
    - What if we have many device services with same key.
    - To review design/implementation at next meeting.
- Up next…
  - Add health/status check of the services to the SMA

        - This will be a call through to the configuration/registry service (Consul)
        - Allows a single point of entry for all EdgeX control plane needs
    - Add system management API to C DS SDK
    - Add system management API to App Services SDK
    - Stretch goal – provide a translation layer (with abstraction) to offer SMA API via other protocol (like LWM2M, SNMP, etc.)
        - At last meeting, merits and options like SNMP, plain REST (as is) or MQTT were discussed as was OneM2M (comment – this is really not used much)
- Long term road map update (see separate doc for this meeting)
    - https://wiki.edgexfoundry.org/download/attachments/329501/EdgeX%20Foundry%20System%20Management%20Roadmap-11-26-18.pdf?version=1&modificationDate=1543285863433&api=v2

New Business

- How is executor distributed?
    - IN Docker-in-docker image
    - IN Snappy – its in the same container with the scripts
    - Is there a volume that SMA has? Use it to host or deposit scripts for start/stop/restart option?
        - Possibly put on roadmap for Fuji and beyond (to be discussed at F2F)
        - Running natively – you would just have startup parameter to look for scripts
        - Some dissention with this opinion. For Docker impl – it should be complete and just work versus set up volume or anything else.