Open Horizon WG Meeting Minutes: 08/26/19

Attendees: Joe, Glen, David (IBM), Ike (Alicon), Timo Honkanen (Vaisala/Finland), Jim, Akram, Trevor (Dell). Attendees who 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 Horizon sub-project
    - We meet again in two weeks, on September 9th.
- Glen had added two files needed to make an official pattern and register it with his organization in a specific Horizon Exchange (input.json and pattern.json). He will confirm that the Pattern works and pulls the containers and can run them.
    - Jim had provided docker-compose.yml
    - Joe successfully did a docker-compose up -d!
    - Recreated input.json (aka metadata to describe the transcription of services).
    - Horizon services instance running, locally.
    - Containers downloaded OK. Issue with volumes…
        - Trevor: Try *docker prune ("docker system prune -a")*
    - Keep an eye on that containers start and do stay up.
- When using docker-compose, how to ensure that EdgeX Foundry is in a proper running state?
- Joe working on a detailed, reproducible set of instructions to capture running the Pattern (in a local end-to-end deployment of Horizon).
- Jim to take the latest Edinburgh 1.0.1 compose file, to get a "minimalist" EdgeX system up and working and seeing data come up through core data (Changed to *device random* for simplicity, and commented out all other services so we can slowly add them back in to the mix as the POC grows).
    - Jim tested it on his end to make sure it works. Goal is to help get us a baseline on known working EdgeX model.
- Joe to merge Jim's commit and tested with *docker-compose -f docker-compose-jpw.yml up* on OSX and Ubuntu 18 on x86.
- The dcmartin repo is presently not available—Investigate other options?
    - No longer on our radar.
    - To use, instead: https://github.com/joewxboy/horizon-edgex
    - Joe shares a demo, to show the execution of instruction in action.
    - Two options for testing in context:
        - on a local instance of the Horizon services that anyone can run,
        - and on a private cloud instance.
            - To prevent false starts and frustrations on the part of the group here, I'd like to test any instructions before we roll them out.
- Based on Jim's new docker-compose-jpw.yml file, Joe to create a corresponding service-jpw.json file.
    - Hope is that it fixes the earlier issue Joe ran into…
    - If so, Joe will then be able to publish his service and corresponding pattern to the (IBM internal) alpha exchange and register my OSX machine's Horizon Agent for the pattern.
- Joe and team continue to work on instructions for setting up a dev exchange instance and/or private cloud exchange instance.
    - In roughly two weeks: Complete troubleshooting issues (with Docker volumes, etc.)
    - Show what telemetry, *device random* serving as the data source.

New Business

- Joe and Glen setting up **Horizon Services** on both a locally-hosted e2edev environment which anyone can replicate, and on an IBM-internal ICP Stack (ICP == IBM Cloud Private).
    - Example currently uses a Service Definition file and a Pattern Definition file for deployment. Now working to migrate Pattern Definition to policy files for automated deployment and ongoing management.
    - The policy files include Node Policy, Service Policy, and Business Policy.
        - Work on an environment that is easily replicable.
        - Have EdgeX running on the Agent
        - Finally, display data (Grafana, etc.)
        - Download repo, copy instructions, get system up and running.
    - 1st three steps done. 4th step being worked.
        - Working on incorporating the config-seed service gracefully.
        - Make the setup be production-ready.
            - Anax (Control system)
            - MiniKube
        - Eventually target OpenShift (RHEL)
    - Main repo for Registering EdgeX Foundry as an Open Horizon Pattern: https://github.com/joewxboy/horizon-edgex
- Current supported environment for **Horizon Services** uses Ubuntu 16 or 18. Plan is to add support for RHEL in the near future. Recommendations for other Linux distributions or OSes?
    - Any other desired environment(s)?
- Current supported environment for **Horizon Agent** is Ubuntu 16, 18, Raspbian Jessie, and OSX Mojave (10.14.6) with RHEL in the near future. Recommendations for others?
    - TBD.
- Joe and team also continue to work on **instructions** for setting up a dev exchange instance and/or private cloud exchange instance.
    - Documented 1st of three steps. Working on the 4th step.
        - Joe referenced the slide "*Basic Telemetry and Simple Inferencing*" (which can be found on the project Wiki (as "OH-EdgeX integration.ppt")
    - Aim is to troubleshoot remaining issues (with Docker volumes, etc.)
        - Complexity in "Policy" aspects. The "Pattern"-based approach simpler, more tractable.
        - David (Martin) working on validating "Policy" area.
        - Glen: Split the deployment into services perhaps.
        - Jim: Request to develop a set of recommendations to polish this POC for consumption. Put this as part of the roadmap, incorporate requirements, etc.
            - Joe: Three types of policies: Have them working with one another in the next few weeks.
        - Ike: Clarity on access policies, such at BT, WiFi, etc.?
            - Glen: Yes.
            - Joe: Specify, for example, what devices are targetable (aka "wakeup policy" per Ike)
        - Jim: When possible to replicate on our end?
            - Joe, Glen: Shortly, possibly in a matter of days.