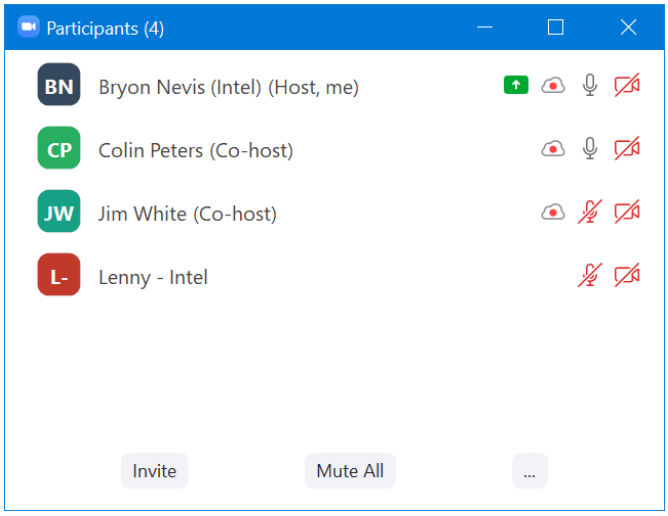
EdgeX Security WG Meeting

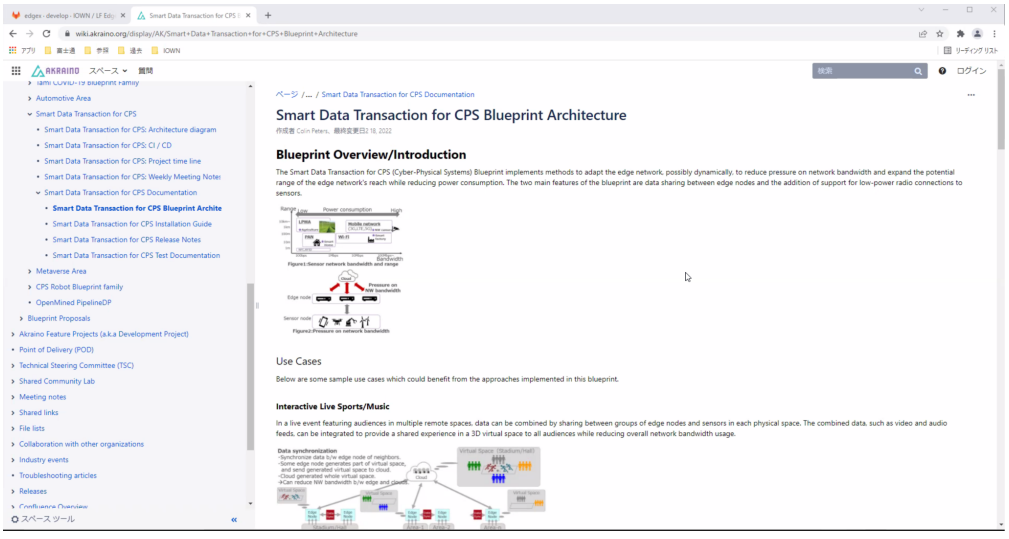
https://wiki.edgexfoundry.org/display/FA/Security+Working+Group

February 23, 2022

Attendees



Agenda

- Presentation: Experiences porting EdgeX to Kubernetes, Colin Peters, Fujitsu
    - Akraino presentation https://wiki.akraino.org/display/AK/Smart+Data+Transaction+for+CPS+Blueprint+Architecture
    - 

- Used strategy of consolidating all containers into a single pod.
- Used Kompose to automate the conversion
  - Sys-mgmt-agent doesn't properly map docker.sock, for example
- Problems occurred when tried to enable security

Workarounds

This folder also contains workarounds for the following issues encountered when building configuration with security enabled:

- Github edgex-go issue #3851
- Github edgex-go issue #3852
- A similar issue with the "kong" volume in the "kong" container hiding the `/usr/local/kong` directory, making Kong unable to start
- An issue with Kong iteself, where the host name "localhost" could not be resolved to an IP address in spite of `/etc/hosts` containing the appropriate `127.0.0.1` value

There is also a workaround for the following issue which does not prevent the pod from initializing, and is a result of our decision to combine all EdgeX containers into a single pod:

- The `security-proxy-setup` container exits on succesful completion, but all other containers are expected to run indefinitely and Kubernetes does not support different restart policies for containers in the same pod, resulting in the `security-proxy-setup` container being restarted constantly

The workarounds are implemented by building three new images based on images used in the original EdgeX Foundry docker-compose file, and using the replacements in our Kubernetes deployment definition. The three containers are built using `Makefile` and `Dockerfile` in the directories below:

- `edgex/security-bootstrapper`
- `edgex/security-secretstore-setup`
- `edgex/kong`

- Kubernetes doesn't support container-specific restart policies. Security-proxy-setup terminating upon success causes problems.
- Series of 3 bugs below that rely on docker volume init semantics:

Almost No Containers Start With Security Enabled (Issue 3851)

When security is enabled the EdgeX Foundry docker-compose file translated into a Kubernetes deployment supplies a `command:` overriding the entrypoint of the container and running a script in `/edgex-init`. The `/edgex-init` volume is shared, and should be initialized by the `security-bootstrapper` container, which exposes its `/edgex-init` path into that volume. However, as described in the github issue, Kubernetes does not treat volumes the same way as docker-compose does, and the files inside the container are not copied out to the shared volume on container startup. This results in containers failing to start as their `command:` cannot be executed.

The workaround is implemented in `edgex/security-bootstrapper/Dockerfile` (and `Dockerfile-arm`) by adding a line to `entrypoint.sh` to copy all the files in `/edgex-init` to `/tmp/edgex-init` as shown below, and changing the mount point of the volume for `security-bootstrapper` to `/tmp/edgex-init`.

```
RUN sed -i -e '2 i cp -Rp /edgex-init/* /tmp/edgex-init' /entrypoint.sh
```

`security-secretstore-setup` Reports "could not read master key shares file" Error (Issue 3852)

As described in the github issue, this problem arises from the `/vault/config` being exposed as a volume, hiding the `/vault/config/assets` directory in the container.

The workaround is implemented in `edgex/security-secretstore-setup`, using a `Dockerfile` which adds commands to create the `assets` directory in `/vault/config` when the container is started (in `entrypoint.sh`) as shown below. This is sufficient since the `/vault/config` directory in the original container only contains an empty `assets` directory and no other data.

```
RUN sed -i -e '2 i mkdir -p /vault/config/assets && chown -Rh 100:1000 /vault/' /usr/local/bin/entrypoint.sh
```

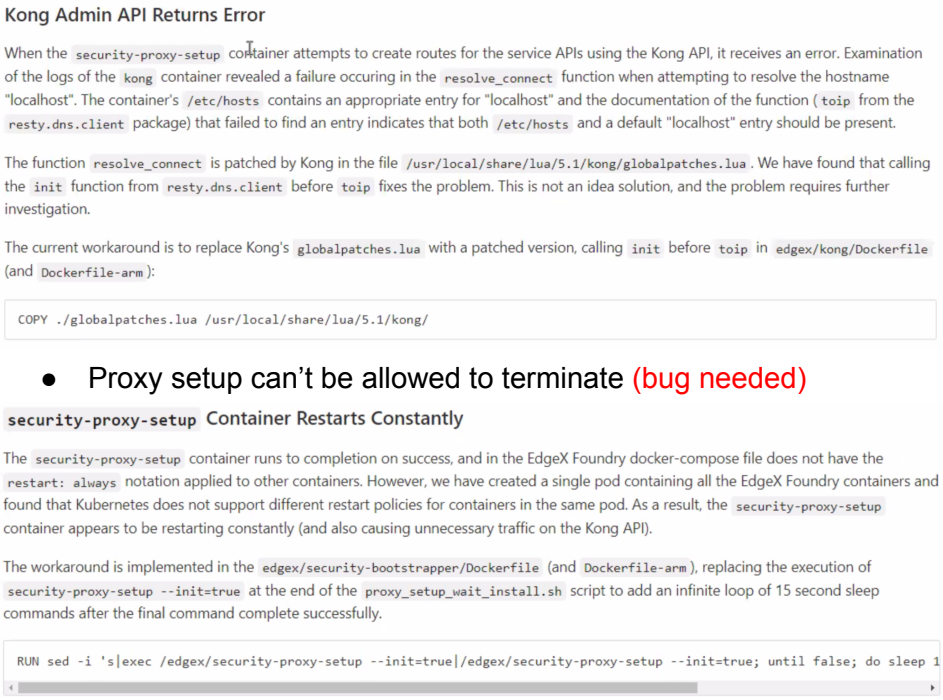
Kong Crashes At Startup With Library Not Found Error

When the `kong` container starts the application itself at the end of the `kong_wait_install.sh` script, the application fails with a missing library error. This is caused by a similar problem to the two issues above. The `/usr/local/kong` directory is exposed as a volume in order that the `security-secretstore-setup` container can supply the `kong.yml` config file for Kong's initialization. However, exposing the `/usr/local/kong` path as a volume hides the `lib` directory underneath it, making it impossible for the application to find and load its libraries.

The workaround is implemented in `edgex/security-bootstrapper/Dockerfile` (and `Dockerfile-arm`) by patching the command in `kong_wait_install.sh` to use `/tmp/kong/kong.yml` instead of `/usr/local/kong/kong.yml` as shown below, and changing the deployment to mount the `kong` volume on `/tmp/kong` in the `kong` container.

```
RUN sed -i -e 's|/usr/local/kong/kong.yml|/tmp/kong/kong.yml|g' ./kong_wait_install.sh
```

- Kong seems to have a localhost resolution bug



Kong Admin API Returns Error

When the `security-proxy-setup` container attempts to create routes for the service APIs using the Kong API, it receives an error. Examination of the logs of the `kong` container revealed a failure occuring in the `resolve_connect` function when attempting to resolve the hostname "localhost". The container's `/etc/hosts` contains an appropriate entry for "localhost" and the documentation of the function (`toip` from the `resty.dns.client` package) that failed to find an entry indicates that both `/etc/hosts` and a default "localhost" entry should be present.

The function `resolve_connect` is patched by Kong in the file `/usr/local/share/lua/5.1/kong/globalpatches.lua`. We have found that calling the `init` function from `resty.dns.client` before `toip` fixes the problem. This is not an idea solution, and the problem requires further investigation.

The current workaround is to replace Kong's `globalpatches.lua` with a patched version, calling `init` before `toip` in `edgex/kong/Dockerfile` (and `Dockerfile-arm`):

```
COPY ./globalpatches.lua /usr/local/share/lua/5.1/kong/
```

- Proxy setup can't be allowed to terminate (bug needed)

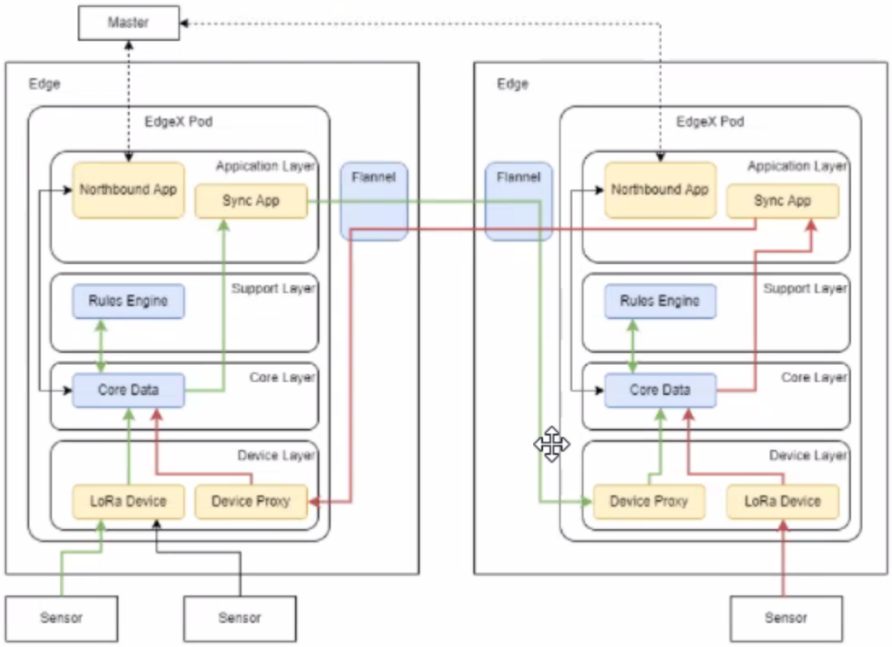
`security-proxy-setup` Container Restarts Constantly

The `security-proxy-setup` container runs to completion on success, and in the EdgeX Foundry docker-compose file does not have the `restart: always` notation applied to other containers. However, we have created a single pod containing all the EdgeX Foundry containers and found that Kubernetes does not support different restart policies for containers in the same pod. As a result, the `security-proxy-setup` container appears to be restarting constantly (and also causing unnecessary traffic on the Kong API).

The workaround is implemented in the `edgex/security-bootstrapper/Dockerfile` (and `Dockerfile-arm`), replacing the execution of `security-proxy-setup --init=true` at the end of the `proxy_setup_wait_install.sh` script to add an infinite loop of 15 second sleep commands after the final command complete successfully.

```
RUN sed -i 's|exec /edgex/security-proxy-setup --init=true|/edgex/security-proxy-setup --init=true; until false; do sleep 1
```

- Discussion: does EdgeX in Kubernetes make sense?
  - Doesn't really fit because of the hardware affinity for device connectivity.
  - Haven't tried Helm chart model yet
  - Akraino blueprint likely to go out as-is
  - "Ansible scripts are a bit clunky"
- Timeline
  - Blueprint "release 6" is in April
  - Next release is 6 months after that
  - Will start investigating Helm chart work
- Questions
  - Ingress controller - no default one with Akraino
  - Storage provider - currently using hostPath for all the volumes
  - Are remote device services sufficient? – Don't know; didn't try that. Do have a need to be able to work on data coming from multiple nodes.



- Using something called a “Device proxy” right now to pass data across nodes.
- (See recording at :52 minutes for more details)

- — CUT HERE DISCUSS THE REST NEXT WEEK —
- Opens
  - ?
- Delayed start service ADR *update*
  - SPIRE server and agent have been integrated into edgex-compose on a branch
  - Have prototype implementation of security-spiffe-token-provider
  - Working on go-mod-secrets code to get secret store token from spiffe-token-provider
- Bin list
  - Nik Huge - present enhancement request for identity at the edge?

Standing Agenda

- Review Security Board
- Securing Consul Board (skip)
- Review CIS docker scan (will skip unless something changes) (click latest run, go to classic, view console output).
  - Last checked: Tue Nov 16 05:36:01 UTC 2021
- Review Snyk (Jenkins) (will skip unless something changes) (Imagelist)
  - https://app.snyk.io/vuln/SNYK-GOLANG-GOLANGORGXTEXTINTERNALLANGUAGE-2400718
    - Affects all of the 2.0.0 containers (CVSS 6.1)
- Review action items from previous week

Action Items

- 7/14/21: Bryon: Update security policy documentation w.r.t. when to use GitHub security advisories to notify users of issues.