PKI Init binary

- Today – it will generate self-signed certificate by default (with option to use customize cert that is imported from pre-defined volume) that is going to be used by Vault and placed in temporary file system.
- Long term – it could generate certs for all services.
- Certs are stored in temporary file system in folder named for the service
- On reboot, new certificates are created.
- Fixes the problem that all services uses the same certificate (that was in Docker container).

How are we protecting master token for Vault?? Is this going to also be handled by PKI Init? Today, the master token for Vault is placed on disk in the clear and used by Vault initialization.

To be determined.

Issue: could we have startup /bootstrapping dependency with this in that the PKI Init has to complete before the other services startup.

- We will probably have to have an artificial delay or use manual kick off of these. TBD???
- All these are in separate containers today – so no way to check dependency easily.
- Use a semaphore like file to indicate that the cert has been created??

Vault is started (in sealed status)

PKI Setup (Secure Setup)

- Creates a certificate for use for Kong.
- Push the cert into Vault (under namespace specified for Kong)
- In future – this should be made more consistent with above and have all certs generated by the same application (PKI Init)

Secret Store Setup (formerly Vault Worker)

- Vault is started in sealed status and this app's job is to unseal Vault. Just a REST API call to Vault with the master key.
- Create and store credentials for MongoDB (Username/password) that other micro services use into Vault.
- Username – static hardcode – same as what is in the mongo init script. Long term it should be configurable. (Core Data, Metadata, Logging, Notifications, Export Client, Scheduler)
  - What about application services??
- Passwords are created differently for each service (may be done different in the future)

Start Kong database (Postgres)

Start Kong Migration

Start Kong (the API Gateway service)

Proxy Setup

- Get cert for Kong from Vault and apply to Kong
- Initialize proxy path for all the micro services (redirects the Kong address to the actual micro service address)

- Configuration TOML file provides the mapping of Kong URL to micros service URL
- Initialize authentication method (JWT or OAuth2)
    - Authorization check only happens at Kong side
    - In future – we may want services to also authorize

Mongo initialization starts

- Needs to get the master token from the temporary volume
- Use that cert to access Vault to then get all Mongo secrets to initialize mongo DB for the other services
- i.e. Pulls secrets from Vault in order to push them into Mongo
- Details TBD

Services start

- Services each need to get their cert from the temporary volume (how – TBD)