New /persist layout for volumes etc

The layout of images, volumes, etc has evolved organically in EVE with things being slightly different for VM images and containers. Due to some new requirements we should create a new, consistent layout.

This layout change will be accompanied by code in the upgradeconverter to preserve existing VM volumes for already deployed applications.

New requirements

1. The new API for volumes means that volumes will be identified by a volume UUID (and a generation counter). To date they are identified by the app instance UUID plus the sha256 of the image used to create them (and a generation/purge counter).
2. By default it makes sense to keep the volumes in an encrypted filesystem/directory, but some applications which need extra performance should be deployable without encryption.
3. The volumes created from containers should be kept in an encrypted partition.
4. While keeping credentials in a VM image or container is a very bad idea, when the image/container is downloaded from a controlled enterprise-local datastore that could be the case. Thus ContentAddressableStorage as well as the directories used by the downloader and verifier to store things on disk/flash should be encrypted.

In addition, we want to remove the differences in location and naming between containers and VM images.

We also want to make it more clear which agent owns particular directories from a defense-in-depth and storage management perspective.

New layout

/persist/vault is the encrypted top-level directory for the parts needing encryption. This will be encrypted using a key sealed under the PCRs in the TPM, thus must not be used to store information which the device needs to access in order to be able to perform remote attestation with the controller.

/persist/clear is an alternative for the parts not needing encryption (currently this is only proposed for volumes where encryption incurs some overhead for running ECOs). Using that requires the addition of a boolean to the EVE API to specify unencrypted storage for the volume.

/persist/unsealed-vault is a future location which will be encrypted using a key stored in the TPM but not sealed under the PCRs. In the future we can move things which are needed during a post-update boot before re-attestation to this unsealed vault, such as /persist/status/nim/DevicePortConfigList/ which keeps the network configuration across device reboots.

Under those three directories we will in principle have sub-directories as follows, but the use of the future /persist/unsealed-vault is TBD.

Volumes

In /persist/vault/volumes/ and /persist/clear/volumes

The naming of the volumes will be <volumeUUID>#<generation counter> for VM and container-based ones.

Thus /persist/img and /persist/runx/pods/prepared will move into this directory

Volumes which should not be encrypted (TBD: we need to add a boolean to the volume config API for that) will be placed in /persist/clear/volumes

Content addressable storage

In /persist/vault/case

TBD: describe naming convention used; I assume we will use the naming convention used by containered

TBD: Will all of /persist/containerd move here?

Temporary persist files during download

All of /persist/vault/downloader will be owned by downloader. volumemgr will be accessing it to measure and report on disk utilization.

The internal structure of it can change over time, but they are likely to be named using Image UUIDs and/or the claimed sha256 of the objects being downloaded.

This replaces /persist/downloads/*/pending. There is probably no reason to keep the object type in the directory names.

Verifier files

The verifier will own (and be the only one with write access to) /persist/vault/verifier/

This includes the files currently undergoing verification (currently in /persist/downloads/*/verifier) and the output which has completed verification (currrently in /persist/downloads/*/verified).

The internal structure of it can change over time, but the undergoing files are likely to be named using the Image UUID and/or the claimed sha256, while the completed files are likely to be named using just the sha256.

Thus this replaces /persist/downloads/*/{verifier,verified). There is probably no reason to keep the object type in the directory names.

Upgradeconverter

At boot upgradeconverter will look for files in /persist/img/ and extract the old and new configuration from /persist/checkpoint/config to get both the volume UUIDs and volumeReg, and the Drive information from the app instance configuration. That is sufficient to determine which volume UUID is used by which app instance, hence convert the filename as part of moving the files from /persist/img to /persist/vault/volumes

TBD: should we do the same for /persist/runx/pods/prepared? Currently these get re-created on each device reboot due to a bug in the code.