/conf as R/O filesystem

Motivation

Over the years we have accumulated quite a bit of R/W state in what, at least initially, was meant to be a R/O filesystem: /conf. As part of our future transition to ZFS and an alternative partitioning scheme in EVE we need to make sure we move all the R/W state out of it.

Proposal

The following files will remain in /conf and be treated as read-only:

- On-boarding certificate and key
  - onboard.cert.pem
  - onboard.key.pem
- Device identity certificate and key
  - device.cert.pem
  - device.key.pem
- Controller location and identity
  - server
  - root-certificate.pem
  - v2tlsbaseroot-certificates.pem
- Seed EVE configuration
  - DevicePortConfig/
  - binary/json full config
  - grub.cfg boot configuration plugin
  - Force-API-V[12]
  - hardwaremodel

The pillar container will keep seeing /config as a R/W location, but unbeknownst to it the actual location of the backing store for it will change to /persist/config (we may even start with a symlink /config /persist/config to being with and gradually update pillar code to directly look into /persist/config). It will be the job of storage-init container to use /conf as a seed content for the /persist/confing PROVIDED that /persist/config doesn't exist.

This scheme will allow us to be able to always get to "factory defaults" by removing all the content accumulated in /persist/conf, rebooting EVE and getting /persist/conf back to its pristine state as recorded in a R/O /conf.

The key improvement, however, is that the format and initial location of the /conf will be completely hidden away from pillar container and will effectively become an implementation detail of storage-init.

Discussion

Most of this change should be pretty transparent to the code running in pillar (once again – the location of /conf as pillar sees it won't even change). The only place where we may see a change will be handling of device identity (as recorded in device.cert.pem and device.key.pem):

- we will now have an additional option of generating device identity during installation (or even baking it into a live image – since our live images are now generated on the fly)
- we are going to introduce a new file called onboarded.with that will contain a copy of the device.cert.pem that the cloud recorded during onboarding - removing this will will initiate a re-onboarding sequence
- it will be up to tpm manager to make sure that the content of the /config/device.cert.pem is in sync with the key stored in the TPM. tpm manager has a full control over the content of device identity files in the new /persist/config