

Feature Roadmap

K-release (Q2-Q3 2024)

Introduction of CAN Bus support

- Support CAN Bus on EVE (+ several CAN controllers)
- Setup CAN interfaces on the host
- Allow VMs to access CAN interfaces from the host (when configured in the device model)

Additional ARM boards

J-release (Q4 2023)

Security

Explore TPM policies to enable secure update and validation of firmware/software in air-gapped environments

Confidential Computing feasibility study

Support/measure chassis intrusion detection hardware switches

Storage

Initial trial of edge clustered storage (leveraging longhorn)

Networking

Multiple LTE/5G modems, APN credentials

Route templates

Real-time OS feasibility study

Configuration service API (for application configuration/patching)

Additional ARM boards

- Introduction of i.MX8M Plus Platform. Supported devices:
 - Advantech EPC-R3720
 - Phytex phyBOARD-Pollux board
- Introduction of NVIDIA Jetson Xavier NX Platform. Supported devices:
 - Lenovo ThinkEdge SE70
 - NVIDIA Jetson Xavier NX developer kit

Additional hardware platform validation tests

Installation

Interactive installer to allow users to modify defaults of EVE installer images on the fly (such as network bootstrapping config, controller URL, etc.).

I-release (Q4 2022 - Q1 2023)

Security

Further leverage TPM to implement best possible security. Also research Arm security enhancement options, such as use of TrustZone, Arm Trusted Firmware, OPTEE-OS.

Storage

Research dis-aggregated container attached storage.

Networking

Advanced network diagnostics.

Zedrouter refactoring (similar to what was done for NIM).

Installation

Interactive installer to allow users to modify defaults of EVE installer images on the fly (such as network bootstrapping config, controller URL, etc.).

Testing/Validation

Improved hardware validation for new platforms running EVE. Better methods to validate device hardware models.

Implement performance testing periodically to evaluate EVE over time.

Some means for interactive application container access and diagnostics?

More participation in daily test review.

Release Engineering

Added automation to reduce EVE qualification testing.

H-release (Q2-Q3 2022)

TODO: Post links to beginning and ending release notes for this period

Security

Enhanced measurement of boot and roots

Storage

ZFS

- Revise current storage health reporting in EVE OS
- Add
 - Reporting multiple disks
 - [S.M.A.R.T](#) reporting
 - Zpool status errors
- Enable reporting in the cloud
- Online pool grow (manual)
 - If more disks added to the pool

Networking

- Completed [NIM](#) refactoring
- Design for Zedrouter refactoring
- [SRIOV support](#)
- [Exposing GPS data](#) to controller and applications (from GPS which is part of the LTE modem)
- [Publishing of cellular info and metrics](#)
- [Eden framework for EVE connectivity testing](#)

Installation

- Initial support for single-use EVE image installer (USB images only)
 - Supports protobuf-formatted [bootstrap config](#) for non-DHCP network installations
 - Enables easier new user installation experience via "auto-onboarding" capability (single-use keys enable hands-off onboarding of EVE to its controller that generated the installer image)

Remote Access

- EdgeView

Testing/Validation

- More comprehensive network testing
- More automation of daily build testing

Release Engineering

- Consistent and comprehensive documenting of bi-weekly release notes
 - Published to LF Edge EVE wiki
 - Published to LF Edge Slack EVE channel
 - Published to LF Edge EVE mailing list
 - Published to LF Edge EVE repository on GitHub
- "Verified" branding
 - DockerHub
 - GitHub

G-release (Q4 2021, Q1 2022)

Storage

1. NVMe/Vhost
 - Prototype
 - Implement creation of data queues
 - Rework Admin queue creation – move away from hardcoded implementation
 - Implement the minimum set of commands required to operate under linux
 - Implement Shadow Queue
 - Make sure works with Windows
 - Harden the prototype
 - Submit RFC patches to the mailing list once Prototype phase is ready
 - Address comments, work on cleaning up hacks
 - Run correctness tests, implement any missing bits an pieces
2. ZFS
 - First official support
 - Detailed performance analysis and implementation to maximize performance
 - Zvol direct-io evaluation
 - Based on Delphix efforts presented on OpenZFS Dev Summit 2021
 - Async DMU / Async CoW
 - Deferring the reads so writes are not blocked

Networking

1. NIM refactoring (design and most of the implementation done in rel. G)
 - Generically [solve the state reconciliation problem](#) (will be applicable not only for network config management)
 - Better separation of concerns (more sub-components inside)
 - Improve robustness, readability and extensibility
2. [VLAN and LAG support](#)
 - Support VLAN+LAG for EVE management and local network instances
3. [Radio silence mode](#)
4. Network performance testing
 - Prepare lab for network performance testing
5. SR-IOV support - design

Testing/Validation

1. Add test suite to validate that a hardware platform supports EVE (with an eye towards future hardware certification)
 - a. Once passed should submit PR with hardware model description
2. Current all the unit tests and half the integration tests are open source; open source remainder of integration tests

Release Engineering

1. Info
 - a. Monthly Video about EVE
 - b. Release notes distribution
2. CI/CD
 - a. Github Actions refactoring
 - b. Implement performance benchmarking CI/CD
 - c. Speed up the GH Actions
 - d. Deploy DataDog CI
 - e. ROL and Github Actions integration
 - f. DockerHub account verification
 - g. Github Account Verification
 - h. Add Github Registry

Security

1. Enable scalable enforcement of TPM PCR values on controller (PCR values should be the same when the hardware, firmware, config, and EVE are the same)
2. Explore Linux Integrity Measurement Architecture (IMA) with the TPM
3. Signed images from the release engineering process including the kernel SHA for the TPM PCRs

F-release (Q2-Q3 2021)

- [Next generation storage architecture](#) (design approved)
- Extract and report GPS coordinates to controller (if device has a cellular modem with GPS)
- Incremental image download (rsync-like) ([proposed](#))
- Better enforcement of volume size limits for the OCI-derived volumes ([proposed](#))
- EVE self-monitor and log memory and disk usage increases - (first cut [done](#) - <https://github.com/lf-edge/eve/pull/2023>)
- EVE report more hardware counters (e.g., disk/SSD S.M.A.R.T counters) ([proposed](#))
- Combine eve, adam, and eden and just call it "eve"? Commands then looks like: "eve pod deploy ...", "eve generate server", ... and maybe after all the config is done, a command like "eve generate image" could be used to create an EVE-OS image for a specific board or hw arch ([proposed](#))
- Expand EVE test coverage to improve overall stability: performance under stress; search for security vulnerabilities; automated code analysis; board farm daily and release testing ([proposed](#))
- Controlling "VM bundles" (see below for a bigger scope around Local Controller) via "run profiles"
 - Controller feature to define the run-profiles, will allow customer to define the set of VMs that form their applications

- Modifiable via MetaData APIs
- User-owned APP responsible for providing a local API endpoint on the box
- "VM bundle" VM version selection
- Hardening USB passthrough
- Hardening GPU passthrough
- Custom CA for HTTPS downloads
- GCP data store
- USB or SAN data store

E-release (Q1 2021)

- New EVE Logging Infrastructure (replace rsyslogd with simpler and more robust logging)
- Make IP geolocation more robust by moving to controller
- Enable deploying K3S workloads (including shared switch network instances to avoid NAT)
- Download EVE as OCI (one layer so far)
- Initial monitoring support for applications (crash, hang, ...)

D-release (Q4 2020)

- Measured Boot and Remote Attestation to the controller
- Encrypted disk vault keys sealed under the TPM PCR values
- End-to-end security even with content inspecting proxies
- Off-line support - verify that applications come up even if there is no connectivity to the controller
- Off-line support - user-settable reset/reboot time if connectivity is lost to controller (to recover from hung network adapters)
- App side-loading/pre-loading by being able to specify content trees independent of volume creation
- Use containerd as Content Addressible Storage for containers, VM and EVE images
- Verify device passthrough for primary video adapter
- Provide an API so Azure IoT edge workload can report status/metrics for the running modules via EVE to controller
- Non-virtio emulation of devices
- Reduce EVE memory footprint (single zedbox process for most microservices); apply C-group controls
- Experimental support for ZFS for the /persist filesystem including encryption

Future ideas

- Support multiple disks using different forms of RAID configurations (might use ZFS)
- Tailscale integration?
- Build EVE as OCI layers (reduces size of EVE update downloads)
- Local Controller
 - EVE toolkit (AKA MetaData server): read-only endpoints, protected by access control, physical access to the system required
 - Extending meta-data server API's
 - provide local access to system metrics
 - provide a local access to change the run-profiles
 - HW Level metrics (IPMI-like): could be part of the above
- CPU affinity
 - Controlling "VM bundles"
- Infrastructure as code (Terraform plugin?)
- Snapshot/differential VM upgrades
- OCI observability
 - coredumps
 - crashkernel
- Sensor I/O virtualization
- Target upstream component for the next release
 - Alpine
 - Linux Kernel
 - Xen
 - Qemu
 - Golang