

California

The California release is the second formal release of EdgeX Foundry. Artifacts for this release carry a 0.6.X version. Key features for the California release include:

- Reverse proxy
- Secure storage
- Go-based microservice replacements for core, supporting and export services
- Improved documentation, documentation management and documentation delivery
- Additional northbound connectors
- Additional blackbox tests
- ARM 64 support
- Upgrade to Consul
- Device Service SDK requirements definition

Release Information

- This release was granted two additional months to deliver, deviating outside of the standard April and October target release schedule, in order to allow time to replace the Java microservices with Go Lang based services. This release was focused on:
- The initial security services were implemented and serve as building blocks to allow additional security features in future releases.
 - A reverse proxy (otherwise known as an API gateway) was implemented to protect the micro service APIs from unauthorized outside requests. It is implemented via Kong (<https://konghq.com/>) and protects the HTTP/S traffic for this release.
 - Secure storage was provided to manage secrets such as API keys, passwords, certificates for EdgeX. The secure store is implemented via Hashicorp Vault (<https://www.vaultproject.io/>).
 - It allows items such as username/password credentials, certificates, secure tokens, etc. to be persisted and protected within EdgeX.
 - These types of "secrets" will allow EdgeX to, for example, encrypt data, make HTTPS calls to the enterprise, or connect EdgeX to a cloud provider in a secure manner.
 - Some initialization bootstrapping was also provided for these services.
- Performance improvement (both in terms of the size and speed of the platform) was a key goal of this release.
 - Microservices written in Go (versus Java) for all export, supporting and core services (note, support-notifications and support-scheduler services are still pending and will be provided with a dot release).
 - New device service SDKs in Go and C will provide the means to create replacement Go or C device services in the next release (Delhi).
 - These new Go based microservices allow EdgeX to start and operate much more quickly (starting in less than a second per service), take up far less memory and CPU (using approximately 99% less memory per service), and be delivered by significantly smaller program executable (the program artifacts are 90% smaller in most cases).
 - These new Go based microservices already allow EdgeX to hit project defined performance targets - that is to be able to easily run on platforms with resources equivalent to those found on a Raspberry Pi 3. The Go bases services allow EdgeX to run on less than 1GB of RAM, run on a 64bit CPU, use less than 32GB of storage, start in less than 1 minute, and have latency from data ingestion to actuation in under 1 second.
 - The code base for the Go based microservices was moved under a single "mono" repository for ease in development and build processes.
- Additions were made to the export client and distribution microservices to allow for new "northbound" connectivity.
 - New connectors were added to allow EdgeX to get sensor data to XMPP, ThingsBoard IoT, and Brightics IoT.
- The documentation and importantly the management and processes of producing end-user documentation was greatly improved..
 - The API and microservice documentation was moved from the EdgeX Wiki to GitHub so that it is updated/maintained/reviewed like code is (through formal pull requests, etc.).
 - This allows the documentation to also be versioned with the code-base.
 - Use of standard documentation tools and processes, allow the documentation to be "built" by the CI processes and released in more user friendly (and portable) fashion. This includes the release of nicely formatted and linked HTML documentation.
 - The released documentation also creates nicely formatted and linked API documentation from the API RAML specifications for each service.
- Additional blackbox tests were created so as to provide API tests around all microservices. The blackbox tests are executed on any code change by continuous integration processes and allow the community to be alerted as soon as a microservice API has been broken by any update.
- Support for ARM 64 was formally added. The EdgeX project now includes an external build environment and tools to create and test (natively) ARM 64 microservice artifacts.
- Consul was upgraded from version 0.7.0 to 1.1.0. Hashicorp Consul is used at the center if the EdgeX registration/configuration service.
- A requirements document was created to define what the device service should accomplish.
 - This requirements document lays out important definitions that will be used when creating the device service SDK for the Delhi release.
 - Device Service SDKs in Go and C are planned for implementation for the Delhi release.

Release Dates

Freeze Date: June 15, 2018

Release Date: June 30, 2018

Release Docker Compose

Version 0.6.0 Release Docker Compose file: <https://github.com/edgexfoundry/developer-scripts/blob/master/compose-files/docker-compose-california-0.6.0.yml>

Version 0.6.1 Release Docker Compose file: <https://github.com/edgexfoundry/developer-scripts/blob/master/compose-files/docker-compose-california-0.6.1.yml>

- Includes fixes for Rules Engine configuration/bootstrapping
- Includes Go base notification micro service (replacing the Java version)