Project Automatos

Project Proposal

Project Proposal - Project Introduction:

Required Information	Responses (Please list N/A if not applicable)
Name of Project	Automatos
Project Description (what it does, why it is valuable, origin and history)	**What Does Automatos Do?** **Automatos** provides complete end-to-end manageability and infrastructure orchestration for all edges across various IoT verticals such as retail, health care, manufacturing, banking and entertainment. It is designed and built using modular microservices-based software stackfocusing on heterogeneous needs of IoT verticals. **Why is it Valuable?** Automatos provides features for orchestrating real time deterministic and real time workloads, small footprint orchestrators and demonstrate the ability to scale. It also provides additional value-adds such as a policy driven configuration engine, security, AI/ML capabilities, telemetry, automation and can interoperate with any Kubernetes distribution. Automatos can be used to deploy Kubernetes clusters in several ways such as: • Virtual cluster • On-premises cluster existing cluster • Dynamic cluster • Hybrid cloud. The value of Automatos to customers comes from: o Tight integration of Intel platform capabilities with the ability to easily deploy and manage both the platforms (manageability) and the workloads (orchestration) based on customer and solution requirements. These capabilities are critical where data collection and data processing must take place at the edge (often due to latency requirements, or cost and time of sending raw data to the cloud and waiting for the processed results to be delivered back to the edge where the activity takes place. o Ease of deployment, resulting in potential reduction in customer TCO over the solution life. o New business opportunities resulting from solutions created with Automatos orchestration and manageability ingredients. These opportunities can drive industry transformation, enable new customer solutions, and engage with new markets. **Origin and History** Automatos was released as open-source software by Intel Corporation in July 2022, based on Intel® Edge Conductor V0.4.0 for LF EDGE Evaluation purposes. The original Intel® Edge Conductor launched in Oct 2021 as an Internal Intel reference platform reflecting the original technologies such as Orchestration and Manageability technologies including Secure Device Onboarding (SDO) by Intel IoT group. With the complex ecosystem needed for success of this product, we decided to open source Automatos to the community in order to drive eco-system adoption, resolve key industry manageability and orchestration automation pain points, and allow the IoT market to grow faster. We believe that open sourcing with a vibrant ecosystem allows Automatos to evolve into a true orchestrator for all edges across various IoT verticals.
Statement on alignment with Foundation Mission Statement	One of the primary objectives of Automatos is to automate, ease and scale Edge deployments and increase TAM on the Edge. To achieve this goal, a cross-industry collaboration of device manufacturers; distributors; systems integrators; cloud service providers and device management software vendors is required to accelerate adoption. The Linux Foundation is the ideal organization to facilitate this collaboration and accelerate adoption of these important technologies.
High level assessment of project synergy with existing projects under LF Edge, including how the project compliments /overlaps with existing projects, and potential ways to harmonize over time. Responses may be included both here and/or in accompanying documentation.	Automatos will accelerate adoption of cluster formation, deployment and workload orchestration into all IoT ecosystems, helping drive the need for all the current projects in the LF EDGE community. Integration with LF EDGE enabled devices could simplify the installation and deployment of current and newly manufactured devices and orchestrate the workload that runs on them.
Link to *current* Code of Conduct	https://lfprojects.org/policies/code-of-conduct/

2 TAC Sponsors, if identified (Sponsors help mentor projects) - See full definition on Project Stages: Definitions and Expectations	TBD
Project license	Apache License 2.0
Source control (GitHub by default)	GitHub
Issue tracker (GitHub by default)	GitHub
External dependencies (including licenses)	

Comments	License
mbedTLS	**Apache License 2.0**
Apache Commons Collections	**Apache License 2.0**
Apache Commons IO	**Apache License 2.0**
Apache Commons Pool	Apache License 2.0
Apache HttpComponents HttpClient	Apache License 2.0
Apache ServiceMix :: Bundles :: spring-aspects	Apache License 2.0
Apache Tomcat	Apache License 2.0
Bouncy Castle PKIX, CMS, EAC, TSP, PKCS, OCSP, CMP, and CRMF APIs	Bouncy Castle
BouncyCastle	Bouncy Castle
CryptoAuthLib	Microchip Technology Inc.
epid-sdk	Intel
FasterXML jackson-core	Apache License 2.0
github.com/Masterminds/sprig/v3 v3.2.2	MIT License
github.com/moby/moby v1.4.2-0.20200203170920-46ec8731fbce	Apache 2.0
github.com/docker/go-connections v0.4.0	Apache 2.0
github.com/go-openapi/errors v0.19.2	Apache 2.0
github.com/go-openapi/strfmt v0.19.5	Apache 2.0
github.com/go-openapi/swag v0.19.5	Apache 2.0
github.com/go-openapi/validate v0.19.8	Apache 2.0
github.com/mitchellh/mapstructure v1.4.1 // indirect	MIT License
github.com/moby/moby v20.10.5+incompatible	Apache 2.0
github.com/onsi/ginkgo v1.11.0	MIT License
github.com/onsi/gomega v1.15.0	MIT License
github.com/pkg/sftp v1.13.1	BSD 2-Clause
github.com/sirupsen/logrus v1.8.1	MIT License
github.com/spf13/cobra v1.1.3	Apache 2.0
golang.org/x/crypt	BSD-3-Clause

golang.org/x/net	BSD-3-Clause
google.golang.org/grpc v1.27.1	Apache 2.0
google.golang.org/protobuf v1.26.0-rc.1	BSD-3-Clause
github.com/fsnotify/fsnotify v1.5.1	BSD-3-Clause
github.com/prashantv/gostub v1.0.0	MIT License
golang.org/x/sys v0.0.0-20210915083310-ed5796bab164	BSD-3-Clause
golang.org/x/tools v0.1.5	BSD-3-Clause
github.com/docker/distribution	Apache 2.0
github.com/go-resty/resty/v2	MIT License
github.com/moby/term	Apache 2.0
github.com/stretchr/testify	MIT License
docker.io/kindest/base:v20210712-e05318fb	Apache 2.0
docker/compose:1.29.2	Apache 2.0
docker/compose:debian-1.29.2	Apache 2.0
goharbor/prepare:v2.3.0	Apache 2.0
quay.io/metal3-io/ironic:master	Apache 2.0
quay.io/metal3-io/ironic-ipa-downloader:master	Apache 2.0
quay.io/metal3-io/keepalived	Apache 2.0
baremetal-operator	Apache 2.0
ip-address-manager	Apache 2.0
cluster-api-provider-metal3	Apache 2.0
kubeadm-bootstrap-controller	Apache 2.0
kubeadm-control-plane-controller	Apache 2.0
cluster-api-controller	Apache 2.0
projects.registry.vmware.com/cluster_api_provider_bringyourownhost/cluster-api-byoh-controller:v0.1.0	Apache 2.0
ghcr.io/kube-vip/kube-vip:v0.3.5	Apache 2.0
k8s.gcr.io/coredns/coredns:v1.8.4	Apache 2.0
k8s.gcr.io/etcd:3.5.0-0	Apache 2.0
k8s.gcr.io/kube-apiserver:v1.22.3	Apache 2.0
k8s.gcr.io/kube-controller-manager:v1.22.3	Apache 2.0
k8s.gcr.io/kube-proxy:v1.22.3	Apache 2.0
k8s.gcr.io/kube-scheduler:v1.22.3	Apache 2.0
gcr.io/kubebuilder/kube-rbac-proxy:v0.8.0	Apache 2.0
quay.io/jetstack/cert-manager-cainjector:v1.6.1	Apache 2.0
quay.io/jetstack/cert-manager-controller:v1.6.1	Apache 2.0
quay.io/jetstack/cert-manager-webhook:v1.6.1	Apache 2.0
clusterctl-linux-amd64	Apache 2.0

rke_linux-amd64	Apache 2.0
kind-linux-amd64	Apache 2.0
byoh-hostagent-linux-amd64	Apache 2.0
imgpkg-linux-amd64	Apache 2.0
clusterctl-linux-amd64	Apache 2.0
Prometheus	Apache 2.0
quay.io/prometheus/node-exporter:v1.1.2	Apache 2.0
k8s.gcr.io/kube-state-metrics/kube-state-metrics:v2.0.0	Apache 2.0
quay.io/prometheus/alertmanager:v0.21.0	Apache 2.0
"jimmidyson/configmap-reload:v0.5.0	Apache 2.0
"prom/pushgateway:v1.3.1	Apache 2.0
quay.io/prometheus/prometheus:v2.26.0	Apache 2.0
multus	Apache 2.0
docker.io/nfvpe/multus:stable	Apache 2.0
docker.io/nfvpe/multus:stable-ppc64le	Apache 2.0
docker.io/nfvpe/multus:stable-arm64v8	Apache 2.0
calico	Apache 2.0
- docker.io/calico/cni:v3.23.1	Apache 2.0
- docker.io/calico/kube-controllers:v3.23.1	Apache 2.0
- docker.io/calico/node:v3.23.1	Apache 2.0
kubevirt-operator	Apache 2.0
kubevirt-cr	Apache 2.0
portainer-ce	MIT License
intel/intel-gpu-initcontainer:0.21.0	Apache 2.0
intel/intel-gpu-plugin:0.21.0	Apache 2.0
nginx-ingress	Apache 2.0
docker.io/jettech/kube-webhook-certgen:v1.5.1	Apache 2.0
k8s.gcr.io/ingress-nginx/controller:v0.47.0@sha256: a1e4efc107be0bb78f32eaec37bef17d7a0c81bec8066cdf2572508d21351d0b	Apache 2.0
kind-nginx-ingress	Apache 2.0
k8s.gcr.io/ingress-nginx/controller:v0.48.1@sha256: e9fb216ace49dfa4a5983b183067e97496e7a8b307d2093f4278cd550c303899	Apache 2.0
docker.io/jettech/kube-webhook-certgen:v1.5.1	Apache 2.0
antrea/antrea-ubuntu:v1.1.0	Apache 2.0
k8s.gcr.io/nfd/node-feature-discovery:v0.10.0	Apache 2.0
Service: rook-ceph- https://charts.rook.io/release/rook-ceph-v1.8.5.tgz	Apache 2.0
docker image:rook/ceph:v1.8.5	Apache 2.0
Service: rook-ceph-cluster	Apache 2.0
Service: Akri	Apache 2.0
Esp	BSD/Intel
RT-linux docker image: - alpine:latest	MIT License
RT-linux bitnami/kubectl:latest	Apache 2.0

	gcr.io/google-samples/hello-app:1.0	Apache 2.0

Release methodology and mechanics	Automatos currently follows a release cadence of approximately 13 weeks, typically with 9 weeks allocated for development, three weeks for test and validation, and one week for final check and release. Critical defects identified in the three-week test & validation phase are resolved and the code base updated to create a final release candidate for the final week of validation. Release artifacts are generated by a fully automated CI system. Integration test and validation includes both automated and manual testing and provides end-to-end testing of the Automatos components running on Intel based platforms.
Names of initial committers, if different from those submitting proposal	TBD
Current number of code contributors to proposed project	20
Current number of organizations contributing to proposed project	1
Briefly describe the project's leadership team and decision-making process	Balaji Ethirajulu (Intel), Sr.Director - SW Product Management, is the “product owner” and is responsible for identifying the feature roadmap for the Automatos project. He collaborates with members of the Automatos ecosystem, LF Edge members, and other industry leaders to identify emerging requirements and features. We anticipate that this process will expand to include others in a Automatos technical steering committee comprised of community contributors and ecosystem stakeholders. Balaji Ethirajulu has contributed to open-source community for over a decade and has held various leadership positions in Linux Foundation projects over the years. Hussein Alayan (Intel) is the Program Manager for Automatos where he is responsible for planning and processes. He has previously contributed to the LF Edge Secure Device Onboard project. We anticipate that Hussein will be an initial maintainer for the Automatos project. Bruce Jones (Intel) is the chief architect for Automatos. He is responsible for translating the feature roadmap into technical requirements and architectural specifications, for maintenance of the Automatos specification, and for the overall security architecture of Automatos We anticipate that he will continue in this role as part of the Automatos Technical Steering Committee. Yong Hu (Intel), Sanjay Mukherjee (Intel), Nicolae Jascanu (Intel) and David Schneider (Intel) are the technical engineering leads for the Automatos project. Yong and Sanjay are responsible for software development. Nicolae is responsible for validation and David is the responsible for oversite of devops and ci /cd activities. We anticipate that they will be initial maintainers for the Automatos project, with responsibility for ensuring contributions are properly and promptly reviewed and approved, and that they will eventually be joined by other contributors as the community of contributors grows. David Cobbley (Intel): Automatos – Engineering Director. Automatos is a complex project comprising several sub-components spanning embedded devices to cloud services. As the community of contributors grows, we anticipate that the governance model will evolve into a core team/sub-team model like the one used by the Rust project as described here: https://github.com /rust-lang/rfcs/blob/master/text/1068-rust-governance.md.
List of project's official communication channels (slack, irc, mailing lists)	As a recently opened open-source project, we plan to work with LF EDGE in setting up communication and net presence (slack, website, social media, etc...)
Link to project's website	As a recently opened open-source project, we plan to work with LF EDGE in setting up communication and net presence (slack, website, social media, etc...)
Links to social media accounts	As a recently opened open-source project, we plan to work with LF EDGE in setting up communication and net presence (slack, website, social media, etc...)
Existing financial sponsorship	Intel Corporation
Infrastructure needs or requests (to include GitHub/Gerrit, CI/CD, Jenkins, Nexus, JIRA, other ...)	· Automatos is moving its continuous integration (CI) infrastructure to Jenkins. If LF EDGE has an alternative solution, we would be interested in learning more about its capabilities and associated costs. · The Automatos plans to maintain both its source and documentation repositories on GitHub. · The Automatos project plans to move its documentation to GitHub Pages. IF LF EDGE has an alternative solution, we would be interested in learning more about its capabilities and associated costs. · The Automatos project would benefit from access to a Jira instance (or equivalent) managed by LF EDGE. The Automatos project would benefit from access to a Slack channel (or equivalent) managed by LF EDGE.
Currently Supported Architecture	x86, x86-64, ARM
Planned Architecture Support	N/A

Project logo in svg format (see https://github.com/lf-edge/lfedge-landscape#logos for guidelines)	As a recently opened open-source project, we plan to work with LF EDGE in setting up communication and net presence (slack, website, social media, etc...)
Trademark status	N/A
Does the project have a Core Infrastructure Initiative security best practices badge? (See: https://bestpractices.coreinfrastructure.org)	No - however, the team is familiar with the Core Infrastructure security badge process and will consider pursuing that badge in the future.
Any additional information the TAC and Board should take into consideration when reviewing your proposal?	No

Project Proposal - Mapping Criteria and Data:

Stage 1: At Large Projects (formerly 'Sandbox')

Criteria	**Data**
2 TAC Sponsors, if identified (Sponsors help mentor projects) - See full definition on Project Stages: Definitions and Expectations	TBD
A presentation at an upcoming meeting of the TAC, in accordance with the project proposal requirements	Yes
The typical IP Policy for Projects under the LF Edge Foundation is Apache 2.0 for Code Contributions, Developer Certificate of Origin (DCO) for new inbound contributions, and Creative Commons Attribution 4.0 International License for Documentation. Projects under outside licenses may be submitted for consideration, subject to review/approval of the TAC and Board.	Yes
Upon acceptance, At Large projects must list their status prominently on website/readme	Yes

*** For existing Projects requesting Stage 2 or Stage 3 consideration, please update the above with the Stage 2 or Stage 3 Mapping criteria, available at Project Stages Mapping: Criteria and Data

Project Proposal - Taxonomy Data:

Functions (Provide, Consume, Facilitate, or N/A; Add context as needed)

Functions	**(Provide, Consume, Facilitate, or N/A; Add context as needed)**
APIs	Provides
Cloud Connectivity	Provide and Facilitate
Container Runtime & Orchestration	Provide and Facilitate
Data Governance	N/A
Data Models	N/A
Device Connectivity	Consumes – HTTP, HTTPs, with architectural support for other connectivity protocols not yet implemented
Filters/Pre-processing	N/A
Logging	Provides
Management UI	Provides (some use cases)
Messaging & Events	Provides

Notifications & Alerts	Provides
Security	Provides
Storage	Provides

Deployment & Industry Verticals (Support, Possible, N/A; Add context as needed)

Deployment Type	**(Support, Possible, N/A; Add context as needed)**
Customer Devices (Edge Nodes)	Support
Customer Premises (DC and Edge Gateways)	Support
Telco Network Edge (MEC and Far-MEC)	Possible
Telco CO & Regional	Possible
Cloud Edge & CDNs	Support
Public Cloud	Support
Private Cloud	Support

Deployment & Industry Verticals (or X; Add context as needed)

Directly applicable Industry/Verticals use cases	**(or X; Add context as needed)**
Automotive / Connected Car	
Chemicals	X
Facilities / Building automation	
Consumer	X
Manufacturing	
Metal & Mining	
Oil & Gas	
Pharma	
Health Care	
Power & Utilities	
Pulp & Paper	
Telco Operators	
Telco/Communications Service Provider (Network Equipment Provider)	
Transportation (asset tracking)	
Supply Chain	
Preventative Maintenance	
Water Utilities	
Security / Surveillance	
Retail / Commerce (physical point of sale with customers)	
Other - Please add if not listed above (please notify TAC-subgroup@lists.lfedge.org when you add one)	

Deployments (static v dynamic, connectivity, physical placement) - (or X; Add context as needed)

Use Cases	**(or X; Add context as needed)**

Gateways (to Cloud, to other placements)	
NFV Infrastructure	
Stationary during their entire usable life / Fixed placement edge constellations / Assume you always have connectivity and you don't need to store & forward.	
Stationary during active periods, but nomadic between activations (e.g., fixed access) / Not always assumed to have connectivity. Don't expect to store & forward.	X
Mobile within a constrained and well-defined space (e.g., in a factory) / Expect to have intermittent connectivity and store & forward.	
Fully mobile (To include: Wearables and Connected Vehicles) / Bursts of connectivity and always store & forward.	

Compute Stack Layers (architecture classification) - (Provide, Require, or N/A; Add context as needed)

Compute Stack Layers	**(Provide, Require, or N/A; Add context as needed)**
APIs	Provide
Applications	Provide
Firmware	Require
Hardware	Require
Orchestration	Provide
OS	Require
VM/Containers	Provide / Provide

Cloud Stack Layers (architecture classification) - (Provide, Require, or N/A; Add context as needed)

Cloud Stack Layers	**(Provide, Require, or N/A; Add context as needed)**
Applications	Yes
Configuration (drive)	Yes
Content (management system)	Yes
IaaS	Yes
PaaS	Yes
Physical Infrastructure	No
SaaS	TBD