
KubeCon
CloudNativeCon
North America 2023


KubeCon | CloudNativeCon

North America 2023

Efficient Edge Computing: Unleashing the Potential of AI/ML with Lightweight Kubernetes

Alex Mevec - Sr. InfraOps Engineer - Lockheed Martin
Ricardo Noriega - Principal SW Engineer - Red Hat

Edge Computing

The Device Edge Ecosystem

not a datacenter

Medical

Automotive

Industrial

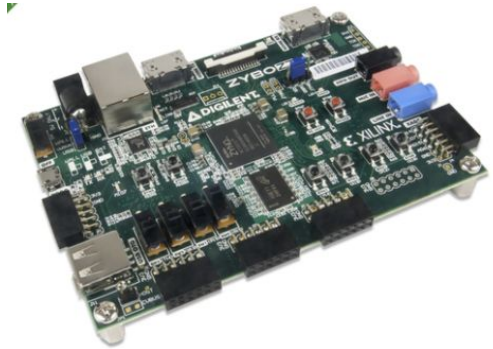
Agricultural

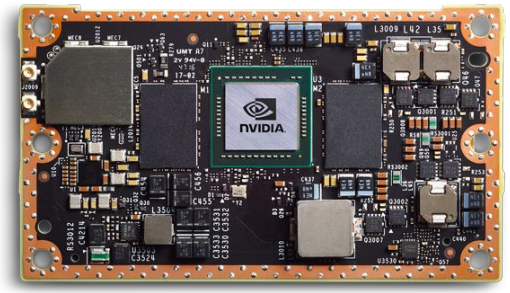
Defense

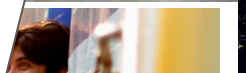
Smart cities and Buildings

https://www.redhat.com/en/technologies/device-edge

Edge Computing







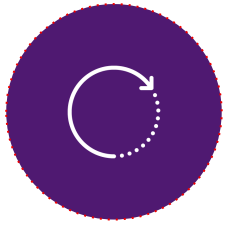
Edge Optimized OS

Quick image generation

Easily create purpose-built OS images optimized for the architectural challenges of the edge.

Edge management

Secure and scale with the benefits of zero-touch provisioning, fleet health visibility, and security remediations throughout the entire lifecycle.



Efficient over-the-air updates

Updates transfer significantly less data and are optimized for remote sites with limited or intermittent connectivity.

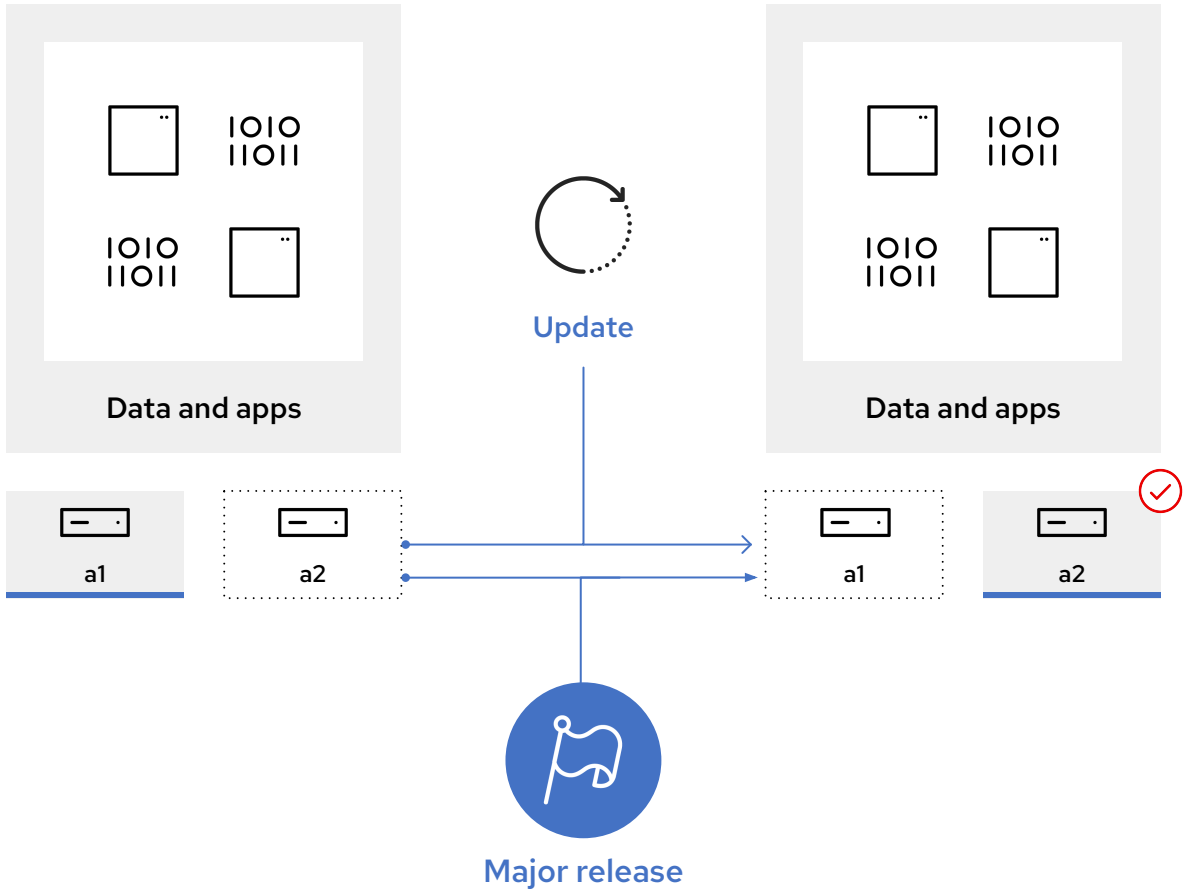
Intelligent rollbacks

Application-specific health checks detect conflicts and automatically reverts to last working OS update, preventing unplanned downtime.

Edge Optimized OS

rpm-ostree

Immutable OS and stateful config and storage



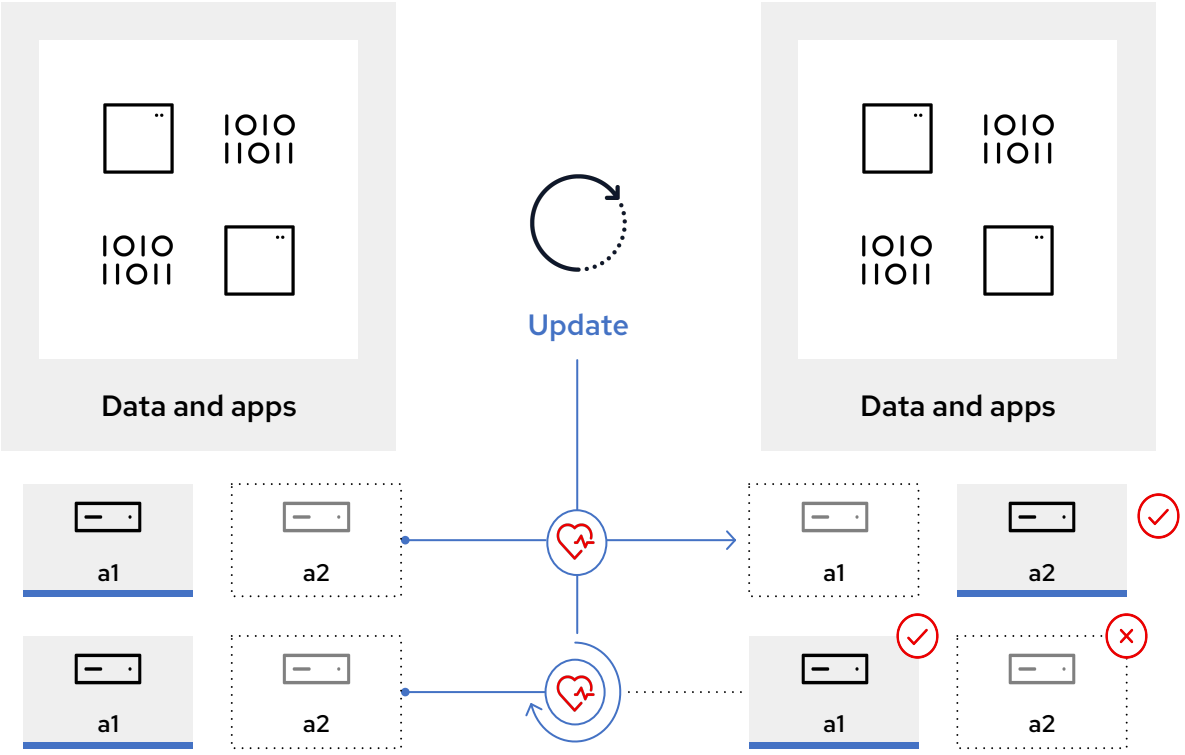
Transactional updates (A → B model)

- OS binaries and libraries (/usr*) are immutable and read-only.
- State (r/w) is maintained in /var and /etc.
- No inbetween state during updates.
- Updates are staged in the background and applied upon reboot.
- Reboots can be scheduled with maintenance windows to ensure the highest possible uptime.

Edge Optimized OS

Intelligent rollbacks: Greenboot

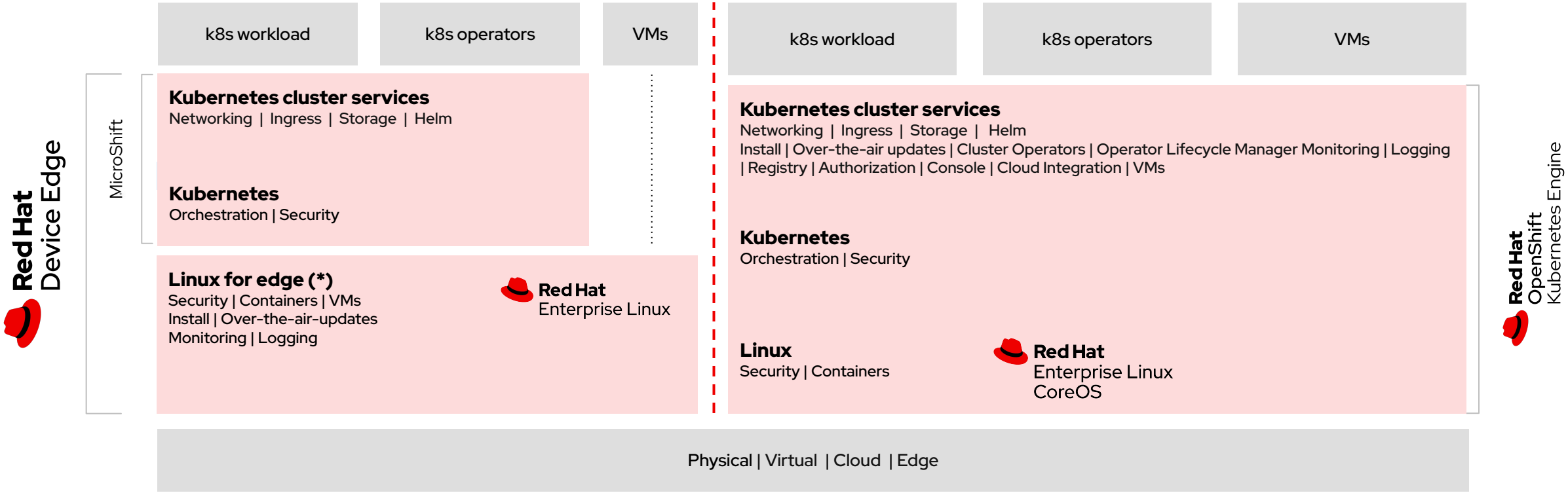
Additional safeguard for application and OS compatibility



Custom health checks can determine if nodes are functioning properly

- Health checks are run during the boot process.
- If checks fail, a counter will track the number of attempts.
- In a failure state, the node will use rpm-ostree to rollback the update.
- Examples can include:
  - Basic name resolution
  - Service or container status or health

MicroShift - a lightweight K8s distro



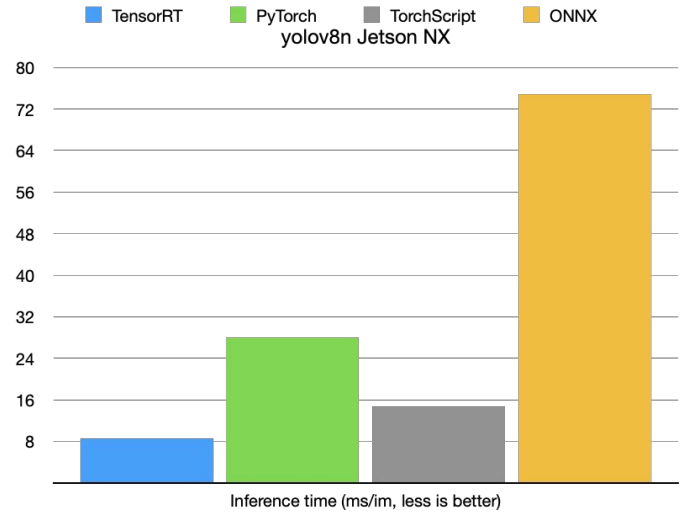
Optimizing Edge AI Models

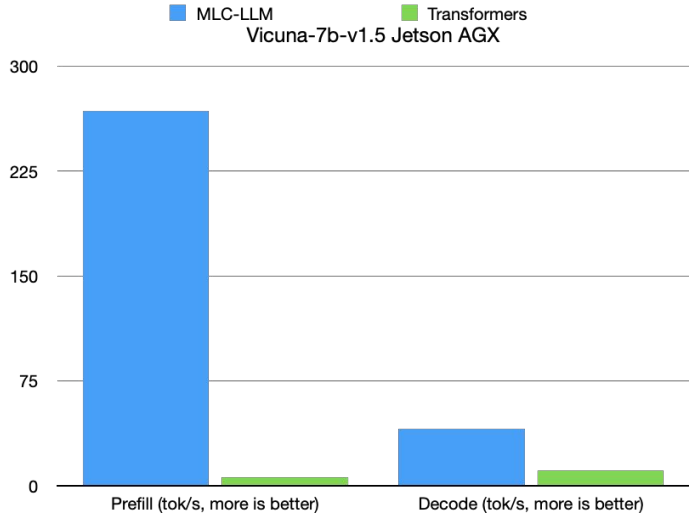
Improvements needed for the edge:

- Resource utilization
- Speed
- Dependencies

Tools used:

- Memory Caching
- TensorRT
  - >95% accuracy
  - >3X speed
- MLC-LLM
  - Speed: >44X prefill & ~4X decode
  - RAM: <10% utilization
  - Note: Tests performed on an AGX due to memory





Edge AI Hosting

What are we looking for?

- Speed
- Model variety
- Dependency management
- Lower resource utilization

Benefits of Kubernetes offerings at the edge:

- Microservices = independent models, independent services
- Containers = portable deployment and training
- Network = extensible connections and scaling
- Probes = self-healing

What tools are we using?

- Protobuf/gRPC
- MicroShift
- Python/Flask

Edge AI Models in Practice

YOLOv8
Image Tracking/Classification
Used to identify various objects in the room to provide LLM context
Optimized with TensorRT

Vicuna-7b-v1.5
Large Language Model
Used to discuss the conference presentation/crowd
Optimized with MLC-LLM

Demo Time

Please scan the QR Code above
to leave feedback on this session