Sharpen the Edge with K3s and Containerized Operating Systems

Rey Lejano, SUSE

whoami

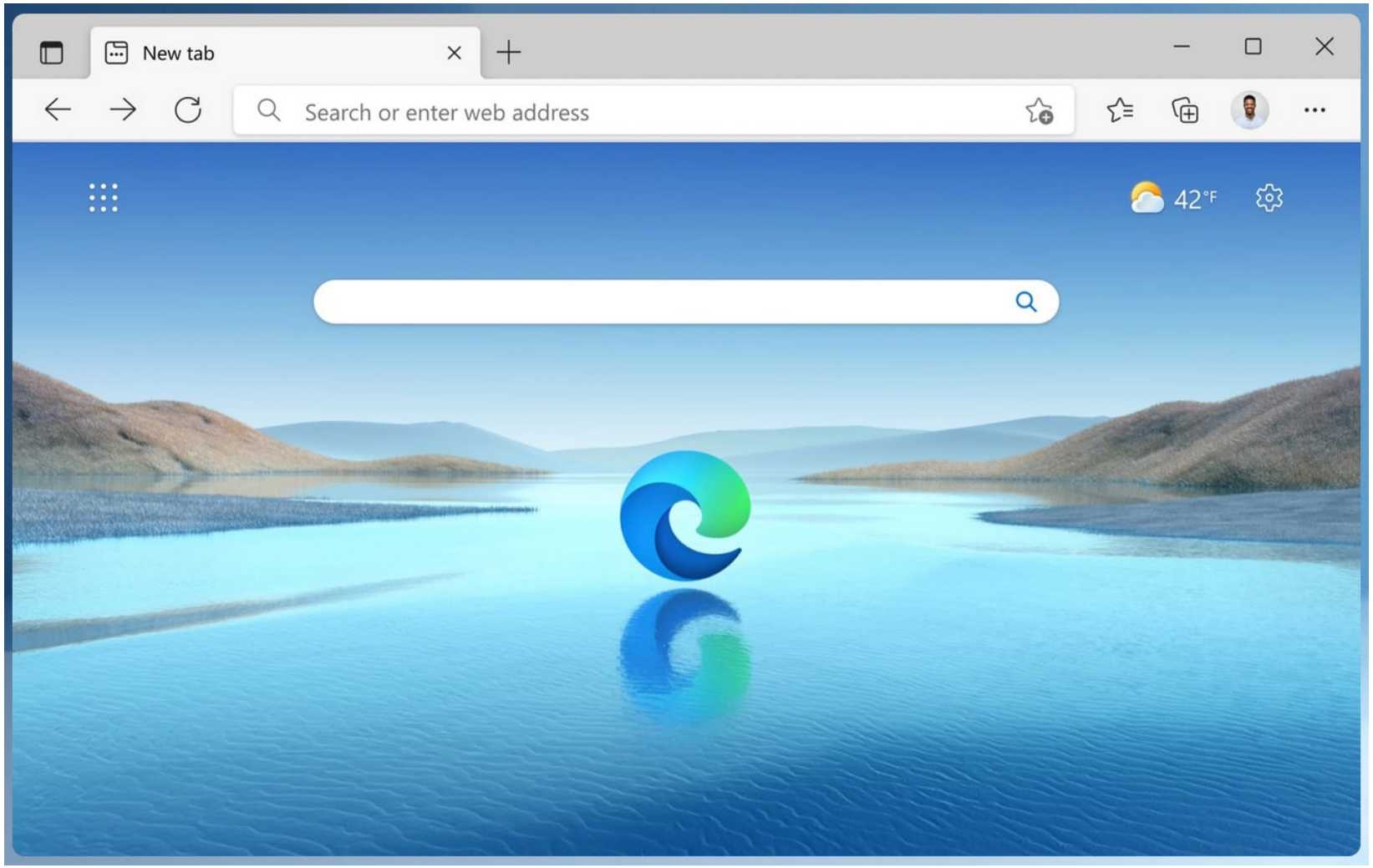
KubeCon CloudNativeCon Europe 2023

Rey Lejano
@reylejano

- Kubernetes Field Engineer, Center of Excellence with SUSE (via Rancher Labs)
- SIG Docs co-chair
- 7 Kubernetes Release Teams
  - v1.23 Release Lead
  - v1.25 Emeritus Adviser
- SIG Security External Audit subproject lead
  - third-party security audit was published on April 19, 2023
  - CNCF blog post: https://www.cncf.io/blog/2023/04/19/new-kubernetes-security-audit-complete-and-open-sourced/

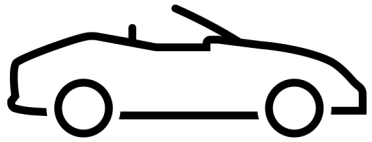
What is the Edge

What is the Edge

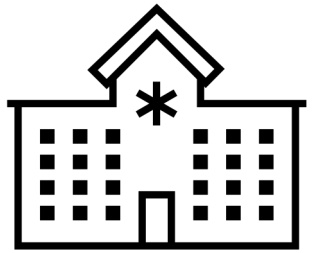


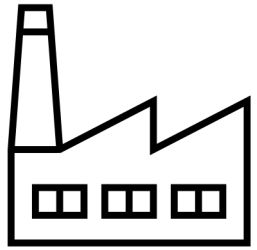
What is the Edge

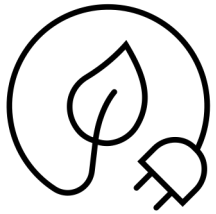
What is the Edge



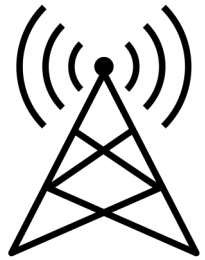
Any remote site/device that processes data and connected by a network



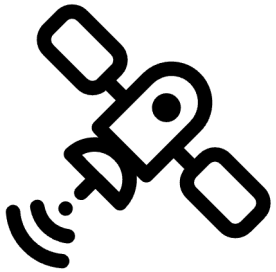




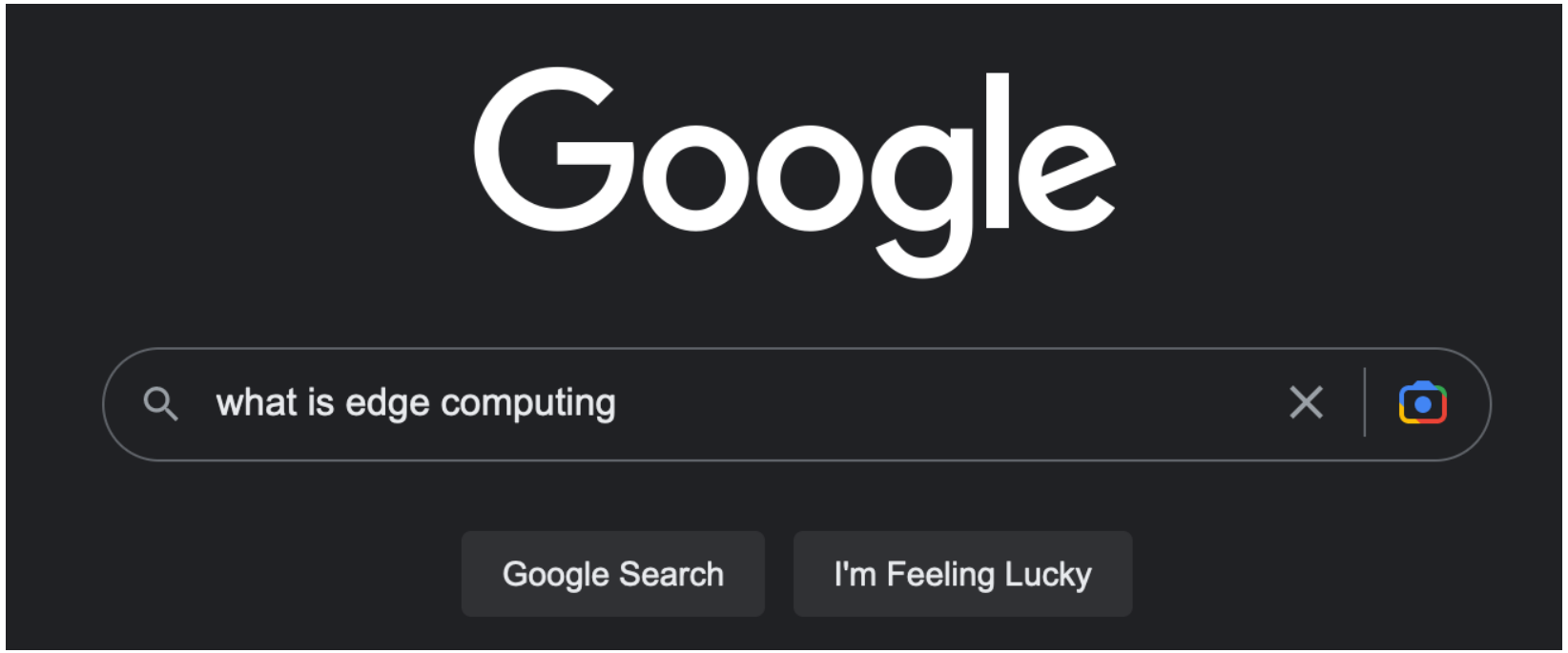
Any place that is not part of an organization’s core data center/network or cloud



Remote device that can sense, inference, and act by itself



What is the Edge



What is the Edge

Wikipedia
Edge computing is a distributed computing paradigm that brings computation and data storage closer to the sources of data. This is expected to improve response times and save bandwidth.

Accenture
Edge computing is an emerging computing paradigm which refers to a range of networks and devices at or near the user. Edge is about processing data closer to where it's being generated, enabling processing at greater speeds and volumes, leading to greater action-led results in real time.

IBM
Edge computing is a distributed computing framework that brings enterprise applications closer to data sources such as IoT devices or local edge servers. This proximity to data at its source can deliver strong business benefits, including faster insights, improved response times and better bandwidth availability.

Microsoft
Edge computing allows devices in remote locations to process data at the "edge" of the network, either by the device or a local server. And when data needs to be processed in the central datacenter, only the most important data is transmitted, thereby minimizing latency.

Linux Foundation State of the Edge https://www.lfedge.org/ :
- The edge is a location, not a thing;
- There are lots of edges, but the edge we care about today is the edge of the last mile network;
- This edge has two sides: an infrastructure edge and a device edge;
- Compute will exist on both sides, working in coordination with the centralized cloud.

Edge Challenges

Resource

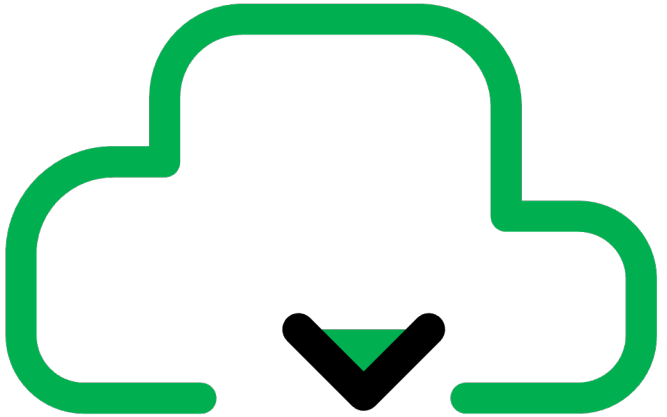
Physical

Provision & Onboard

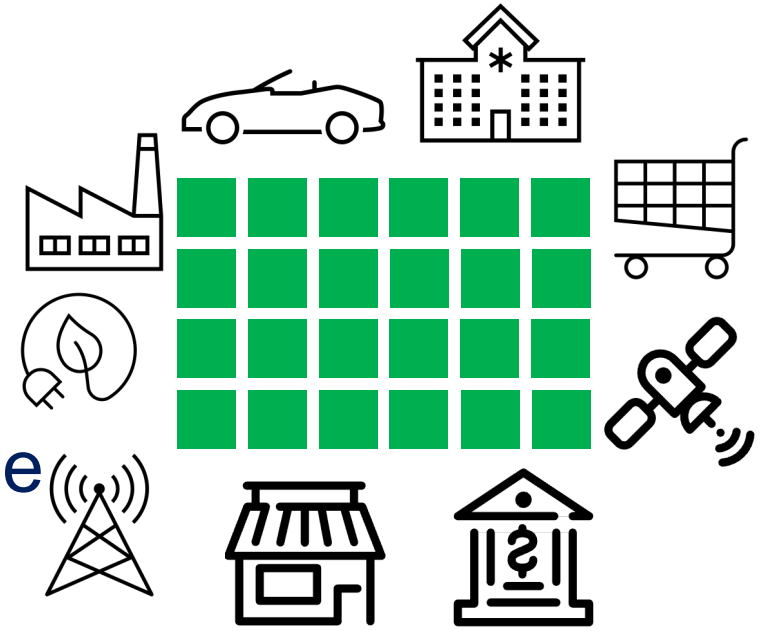
Patch/Upgrade

Born in the Cloud, Works at the Edge

Is it the cloud or edge (it's both)



Benefits of the cloud lead to edge deployments
- Automated pipelines
- GitOps to deploy/manage applications/microservices across clusters
- Infrastructure as code
- Delivering consistent updates



DevOps, containers, and Kubernetes to the Edge

Solving the Edge Puzzle

Kubernetes Distro

OS

Operational Efficiency

Cloud Native with Edge Infra

Kubernetes Distro - K3S

K3S

K3S is a CNCF project (sandbox)

https://github.com/k3s-io/k3s
https://docs.k3s.io/

Packaged as a single binary

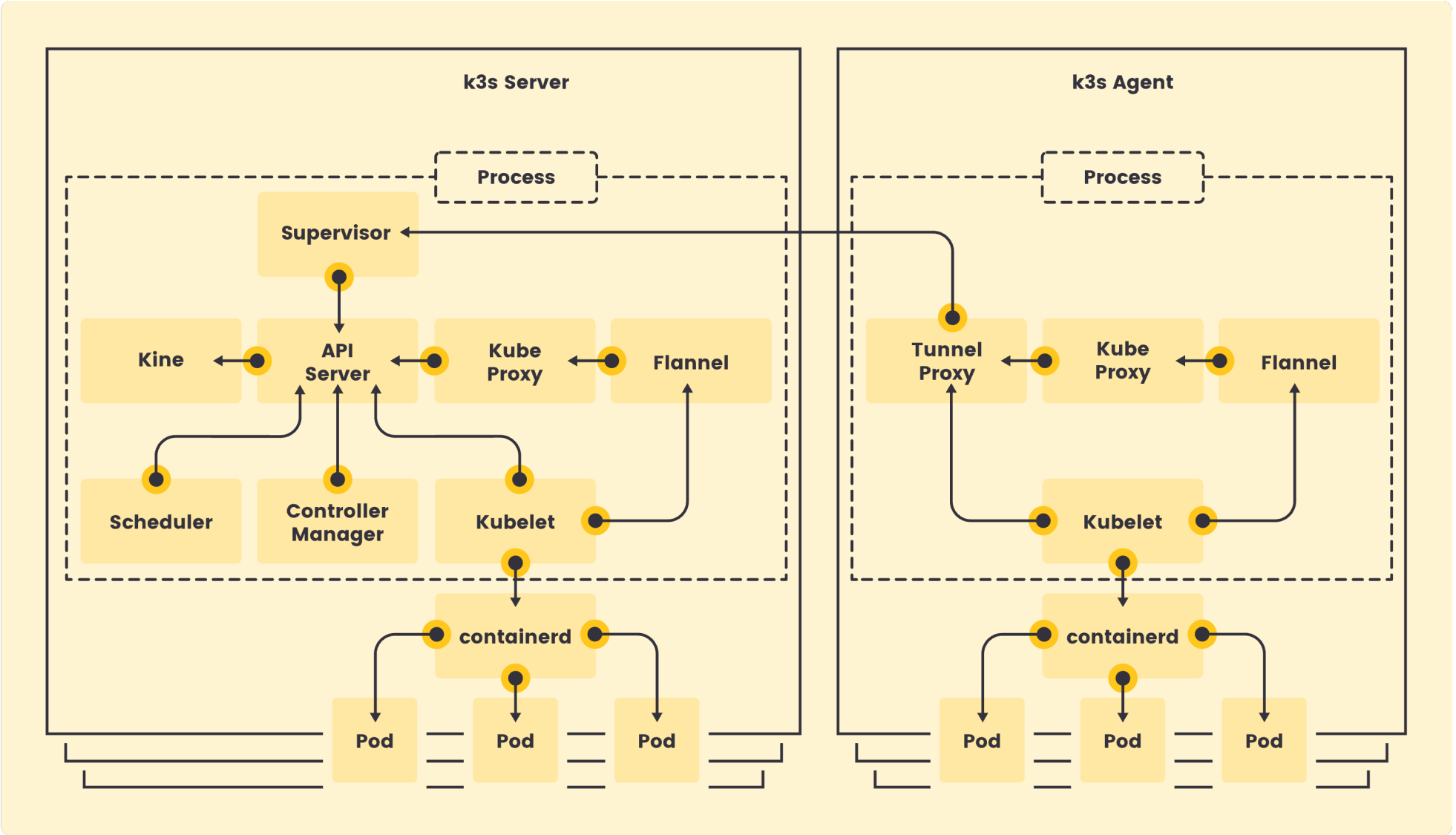
included

- Containerd and runc
- Flannel for CNI
- CoreDNS
- Metrics Server
- Local storage provider
- service load balancer (klipper-lb)
- Helm controller
- Traefik ingress controller
- Kube-router - network policy controller
- Kine is an etcdshim that translates etcd API to SQLite (default), Postgres, MySQL, NATS Jetstream

One-line install:

```
curl -sfL https://get.k3s.io | K3S_KUBECONFIG_MODE=0644 sh -
```

K3S


k3s Server
Process
Supervisor
Kine
API Server
Kube Proxy
Flannel
Scheduler
Controller Manager
Kubelet
containerd
Pod
Pod
Pod
k3s Agent
Process
Tunnel Proxy
Kube Proxy
Flannel
Kubelet
containerd
Pod
Pod
Pod

Containerized OS

OS Container Images

opensuse/tumbleweed ☆

By opensuse • Updated 2 hours ago

Official openSUSE Tumbleweed images

Image

SLE BCI 15 SP4 Base Container Image

bci/bci-base

ubuntu DOCKER OFFICIAL IMAGE • 1B+ • ☆10K+

Ubuntu is a Debian-based Linux operating system based on free software.

alpine DOCKER OFFICIAL IMAGE • 1B+ • ☆9.9K

A minimal Docker image based on Alpine Linux with a complete package index and only 5 MB in size!

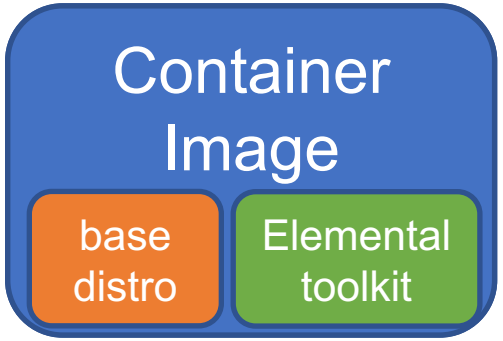
Bootable Containerized OS

How do we make a bootable containerized OS?

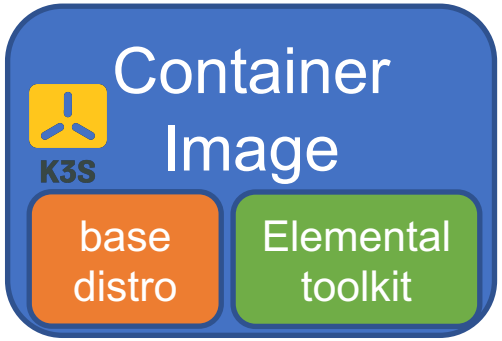
cOS toolkit to build a containerOS
Now called Elemental toolkit
https://github.com/rancher/elemental-toolkit/
https://rancher.github.io/elemental-toolkit/docs

Uses Luet (container based package manager)
https://github.com/mudler/luet

Elemental is a software stack to enable centralized, full cloud-native OS management with Kubernetes
https://elemental.docs.rancher.com/



Making a Containerized OS



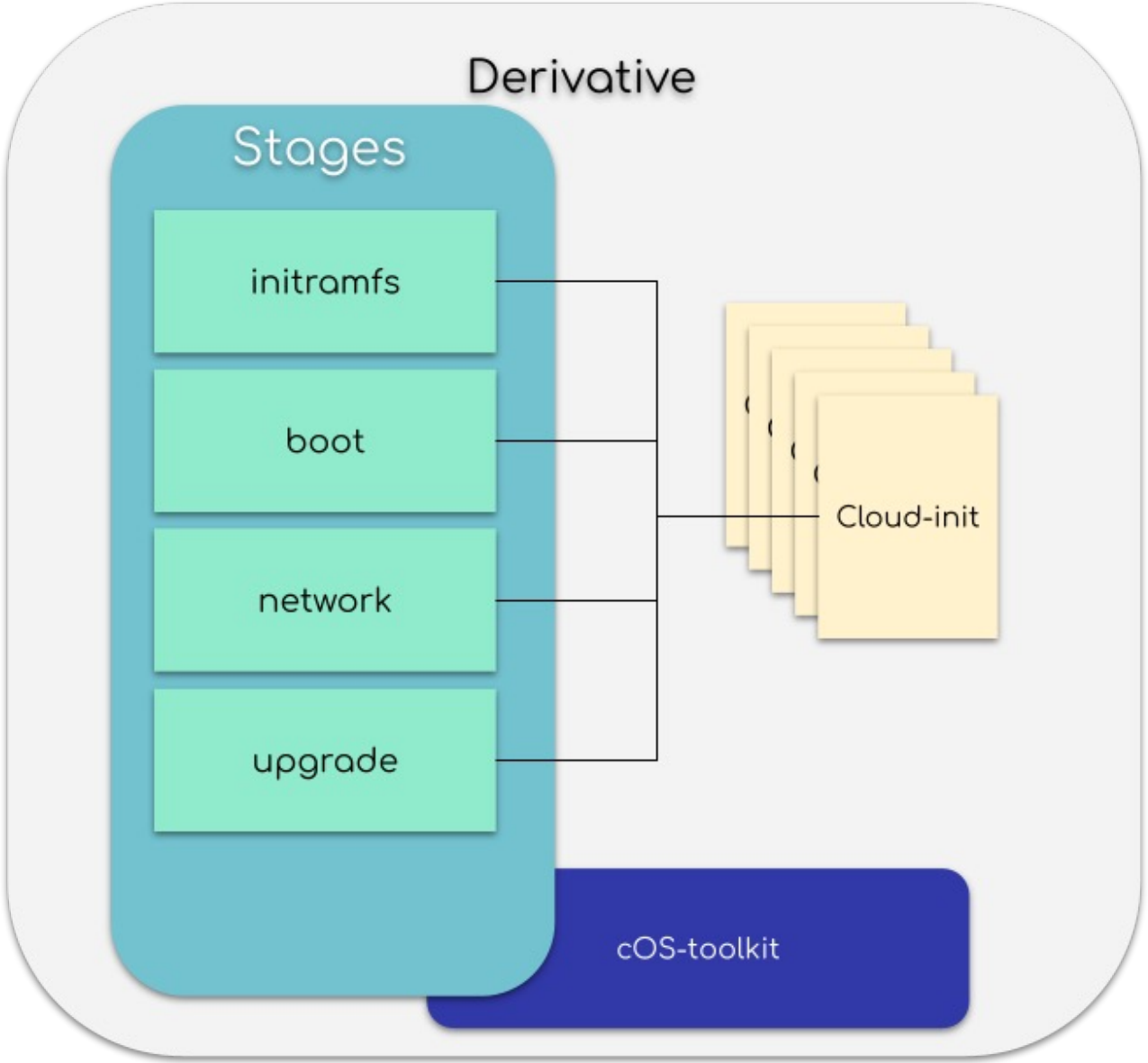
Elemental Toolkit

- Includes the runtime and buildtime framework to boot containers in VMs, Cloud, Baremetal
- Container image can be booted as-is or used to create an installation medium: iso, raw image, ova, cloud, ipxe, vagrant, qcow2
- Additional customizations via cloud-init
- A/B upgrades, immutable systems
- Can embed K3S

Elemental toolkit includes:

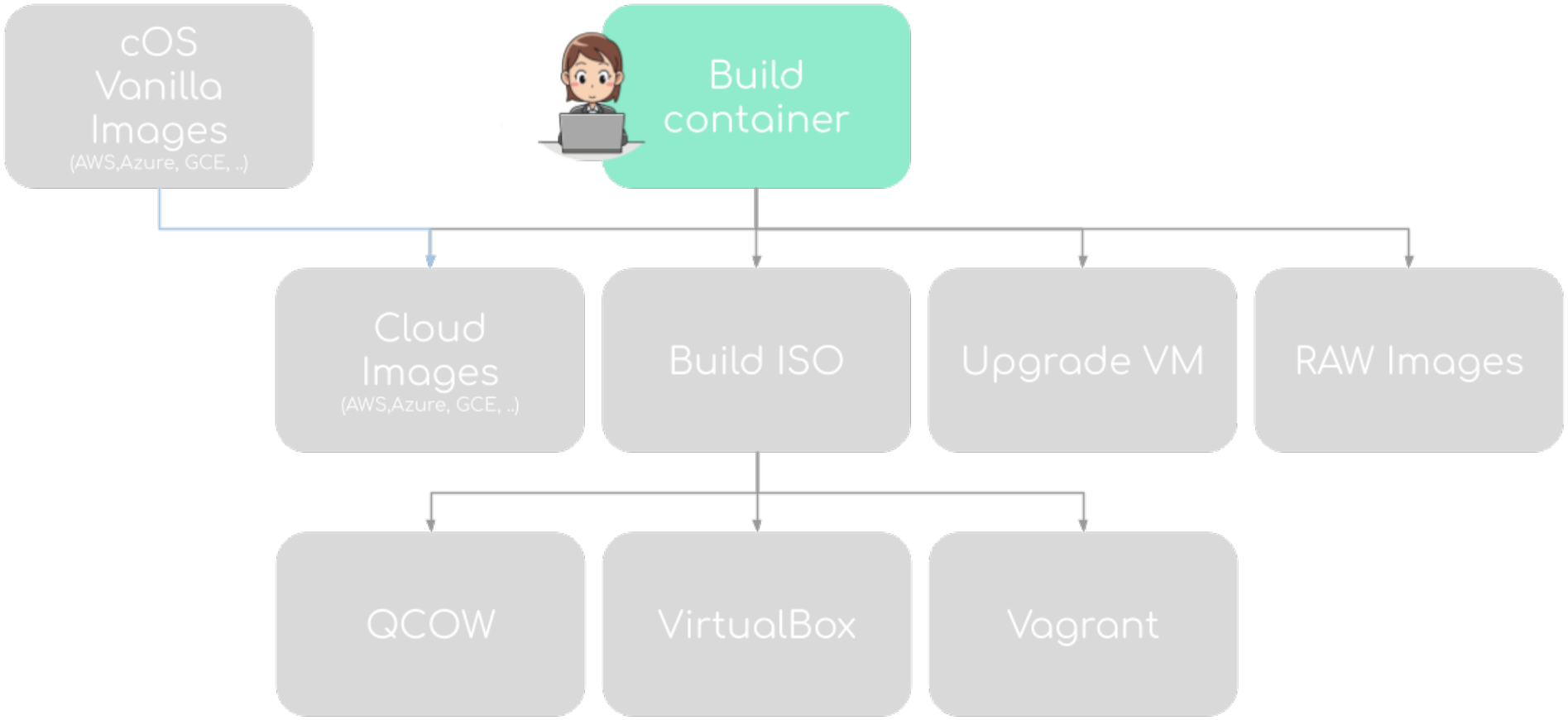
- GRUB and Dracut for loading and automating the boot process

Configuring Containerized OS



Cloud-init files in /system/oem, /oem and /usr/local/oem are applied in 5 different phases: boot, network, fs, initramfs and reconcile

Making a Containerized OS



Making a Containerized OS

Dockerfile

```
# Let's copy over luet from official images.
# This version will be used to bootstrap luet itself and Elemental internal components
ARG LUET_VERSION=0.32.0
FROM quay.io/luet/base:$LUET_VERSION AS luet

FROM registry.suse.com/bci/bci-minimal:15.4
ARG K3S_VERSION=v1.24.10+k3s1
ARG ARCH=amd64
ENV ARCH=${ARCH}
ENV LUET_NOLOCK=true


# Copy the luet config file pointing to the upgrade repository
COPY repositories.yaml /etc/luet/luet.yaml

# Copy luet from the official images
COPY --from=luet /usr/bin/luet /usr/bin/luet
RUN luet install -y \
      toolchain/yip \
      toolchain/luet \
      utils/installer \
      system/cos-setup \
      system/immutable-rootfs \
      system/grub2-config \
      system/base-dracut-modules

# Install k3s server/agent
ENV INSTALL_K3S_VERSION=${K3S_VERSION}
RUN curl -sfL https://get.k3s.io > installer.sh && \
    INSTALL_K3S_SKIP_START="true" INSTALL_K3S_SKIP_ENABLE="true" sh installer.sh && \
    INSTALL_K3S_SKIP_START="true" INSTALL_K3S_SKIP_ENABLE="true" sh installer.sh agent && \
    rm -rf installer.sh

## System layout

# Required by k3s etc.
RUN mkdir /usr/libexec && touch /usr/libexec/.keep

# Copy custom files
# COPY files/ /

# Copy cloud-init default configuration
COPY cloud-init.yaml /system/oem/

# Generate initrd
RUN mkinitrd

# OS level configuration
RUN echo "VERSION=999" > /etc/os-release
RUN echo "GRUB_ENTRY_NAME=derivative" >> /etc/os-release
RUN echo "welcome to our derivative" >> /etc/issue.d/01-derivative

# Copy cloud-init default configuration
COPY cloud-init.yaml /system/oem/
```

Making a Containerized OS

deriative/cloud-init.yaml

```
# See https://rancher.github.io/elemental-toolkit/docs/reference/cloud_init/ for a full
syntax reference
name: "Default settings"
stages:
  initramfs:
    # Setup default hostname
    - name: "Branding"
      hostname: "derivative"
    # Setup an admin group with sudo access
    - name: "Setup groups"
      ensure_entities:
      - entity: |
          kind: "group"
          group_name: "admin"
          password: "x"
          gid: 900
    # Setup network - openSUSE specific
    - name: "Network setup"
      files:
      - path: /etc/sysconfig/network/ifcfg-eth0
        content: |
              BOOTPROTO='dhcp'
              STARTMODE='onboot'
        permissions: 0600
        owner: 0
        group: 0
    # Setup a custom user
    - name: "Setup users"
      users:
```

```
# Replace the default user name here and settings
        rey:
          # Comment passwd for no password
          passwd: ”rey"
          shell: /bin/bash
          homedir: "/home/rey"
          groups:
          - "admin"
        #authorized_keys:
        # Replace here with your ssh keys
        # rey:
        # - ssh-rsa ....
      # Setup sudo
      - name: "Setup sudo"
        files:
        - path: "/etc/sudoers"
          owner: 0
          group: 0
          permsisions: 0600
          content: |
            Defaults always_set_home
            Defaults secure_path="/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/bin:/usr/local/sbin"
            Defaults env_reset
            Defaults env_keep = "LANG LC_ADDRESS LC_CTYPE LC_COLLATE LC_IDENTIFICATION
LC_MEASUREMENT LC_MESSAGES LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
LC_TELEPHONE LC_ATIME LC_ALL LANGUAGE LINGUAS XDG_SESSION_COOKIE"
            Defaults !insults
            root ALL=(ALL) ALL
            %admin ALL=(ALL) NOPASSWD: ALL
            @includedir /etc/sudoers.d
        commands:
        - passwd -l root
```

Making a Containerized OS

deriative/cloud-init.yaml

```
# Setup persistency so k3s works properly
# See also: https://rancher.github.io/elemental-toolkit/docs/reference/immutable_rootfs/#configuration-with-an-environment-file
rootfs.after:
 - name: "Immutable Layout configuration"
   environment_file: /run/cos/cos-layout.env
   environment:
     VOLUMES: "LABEL=COS_OEM:/oem LABEL=COS_PERSISTENT:/usr/local"
     OVERLAY: "tmpfs:25%"
     RW_PATHS: "/var /etc /srv"
     PERSISTENT_STATE_PATHS: >-
       /etc/systemd
       /etc/rancher
       /etc/ssh
       /etc/iscsi
       /etc/cni
       /home
       /opt
       /root
       /usr/libexec
       /var/log
       /var/lib/rancher
       /var/lib/kubelet
       /var/lib/NetworkManager
       /var/lib/longhorn
       /var/lib/cni
     PERSISTENT_STATE_BIND: "true"
# Finally, let's start k3s when network is available, and download the SSH key from github for the rey user
network:
  - name: "Start k3s"
    systemctl:
      start:
      - k3s
  - authorized_keys:
  # Replace here with your ssh keys or github handle
     rey:
     - github:reylejano
```

Making a Containerized OS

deriative/repositories.yaml

```
logging:
  color: false
  enable_emoji: false
general:
   debug: false
   spinner_charset: 9
repositories:
- name: "mylinuxderivative "
  description: " mylinuxderivative"
  type: "docker"
  enable: true
  cached: true
  priority: 1
  verify: false
  urls:
  - "quay.io/costoolkit/releases-green"
```

Making a Containerized OS

Creating the container image of a bootable containerized OS

```
docker build -t mylinuxderivative:1.2
```

```
... ---> Running in a9c33b42f567 Removing intermediate
container a9c33b42f567 ---> 8e83191d29df Step 19/19 :
COPY cloud-init.yaml /system/oem/ ---> 38cc4c8b173a
Successfully built 38cc4c8b173a Successfully tagged
mylinuxderivative:1.2
```

Making a Containerized OS

Creating the ISO of a bootable containerized OS

deriative/iso.yaml

```
packages:
  uefi:
  - live/grub2-efi-image
  isoimage:
  - live/grub2
  - live/grub2-efi-image

boot_file: "boot/x86_64/loader/eltorito.img"
boot_catalog: "boot/x86_64/boot.catalog"
isohybrid_mbr: "boot/x86_64/loader/boot_hybrid.img"

initramfs:
  kernel_file: "vmlinuz"
  rootfs_file: "initrd"

image_prefix: " mylinuxderivative-1."
image_date: true
label: "COS_LIVE"

luet:
  repositories:
  - name: Elemental
    enable: true
    urls:
      - quay.io/costoolkit/green
    type: docker
```

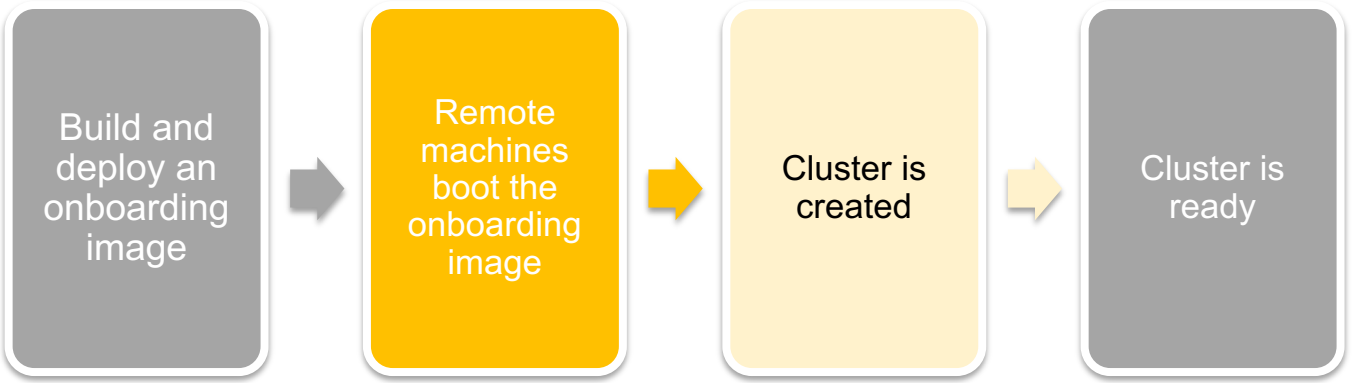
Making a Containerized OS

Creating the ISO of a bootable containerized OS

```
docker run -v $PWD:/cOS -v /var/run:/var/run --entrypoint /usr/bin/luet-makeiso -ti --rm
quay.io/costoolkit/toolchain ./iso.yaml --image mylinuxderivative:1.2
....
INFO[0114] Copying BIOS kernels
INFO[0114] Create squashfs
Parallel mksquashfs: Using 8 processors
Creating 4.0 filesystem on /tmp/luet-geniso4082786464/tempISO/rootfs.squashfs, block size 1048576.
....
INFO[0247] 🍹 Generate ISO derivative-0.20210909.iso
xorriso 1.4.6 : RockRidge filesystem manipulator, libburnia project.

Drive current: -outdev 'stdio:derivative-0.20210909.iso'
Media current: stdio file, overwriteable
Media status : is blank
Media summary: 0 sessions, 0 data blocks, 0 data,  448g free
Added to ISO image: directory '/'='/tmp/luet-geniso4082786464/tempISO'
xorriso : UPDATE : 599 files added in 1 seconds
xorriso : UPDATE : 599 files added in 1 seconds
xorriso : NOTE : Copying to System Area: 512 bytes from file '/tmp/luet-
geniso4082786464/tempISO/boot/x86_64/loader/boot_hybrid.img'
xorriso : WARNING : Boot image load size exceeds 65535 blocks of 512 bytes. Will record 0 in El Torito to
extend ESP to end-of-medium.
libisofs: NOTE : Aligned image size to cylinder size by 137 blocks
xorriso : UPDATE :  12.35% done
xorriso : UPDATE :  42.73% done
ISO image produced: 282624 sectors
Written to medium : 282624 sectors at LBA 0
Writing to 'stdio: mylinuxderivative-1.20210909.iso' completed successfully.
```

Provisioning / Onboarding Flow

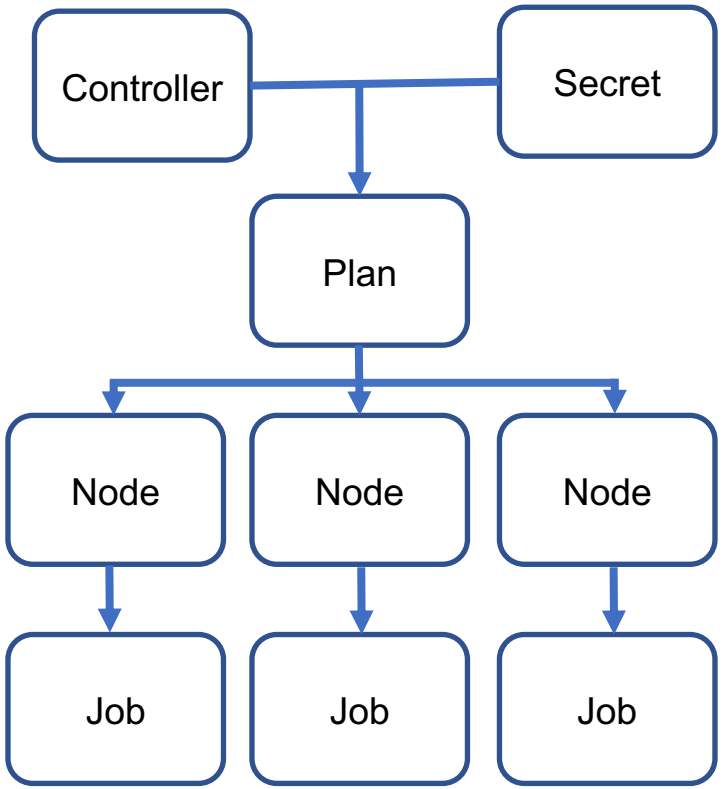


Patching and Upgrades

System Update Controller - Kubernetes controller that can upgrade the node
https://github.com/rancher/system-upgrade-controller
CNCF Webinar: https://www.cncf.io/webinars/declarative-host-upgrades-from-within-kubernetes/

Very privileged container ← pod ← job
- Namespaces:
host IPC, NET, and PID
- Capability: CAP_SYS_BOOT
- host root file-system mounted at /host (read/write)

Plan CRD: upgrade policies and requirements



Patching and Upgrading


```
---
apiVersion: upgrade.cattle.io/v1
kind: Plan
metadata:
  name: elemental-upgrade
  namespace: system-upgrade
  labels:
    k3s-upgrade: server
spec:
  concurrency: 1
  version:  fleet-sample # Image tag
  nodeSelector:
    matchExpressions:
      - {key: k3s.io/hostname, operator: Exists}
  serviceAccountName: system-upgrade
  cordon: true
#  drain:
#    force: true
  upgrade:
    image: quay.io/costoolkit/mylinuxderivative:1.2 # Image upgrade reference
    command:
    - "/usr/sbin/suc-upgrade"
```

Upgrade via:

- System Upgrade Controller
- ISO
- Elemental CLI

On install: 2 .img image files in `COS_STATE` partition:

`/cos/active.img`
`/cos/passive.img`

OS Management Flow

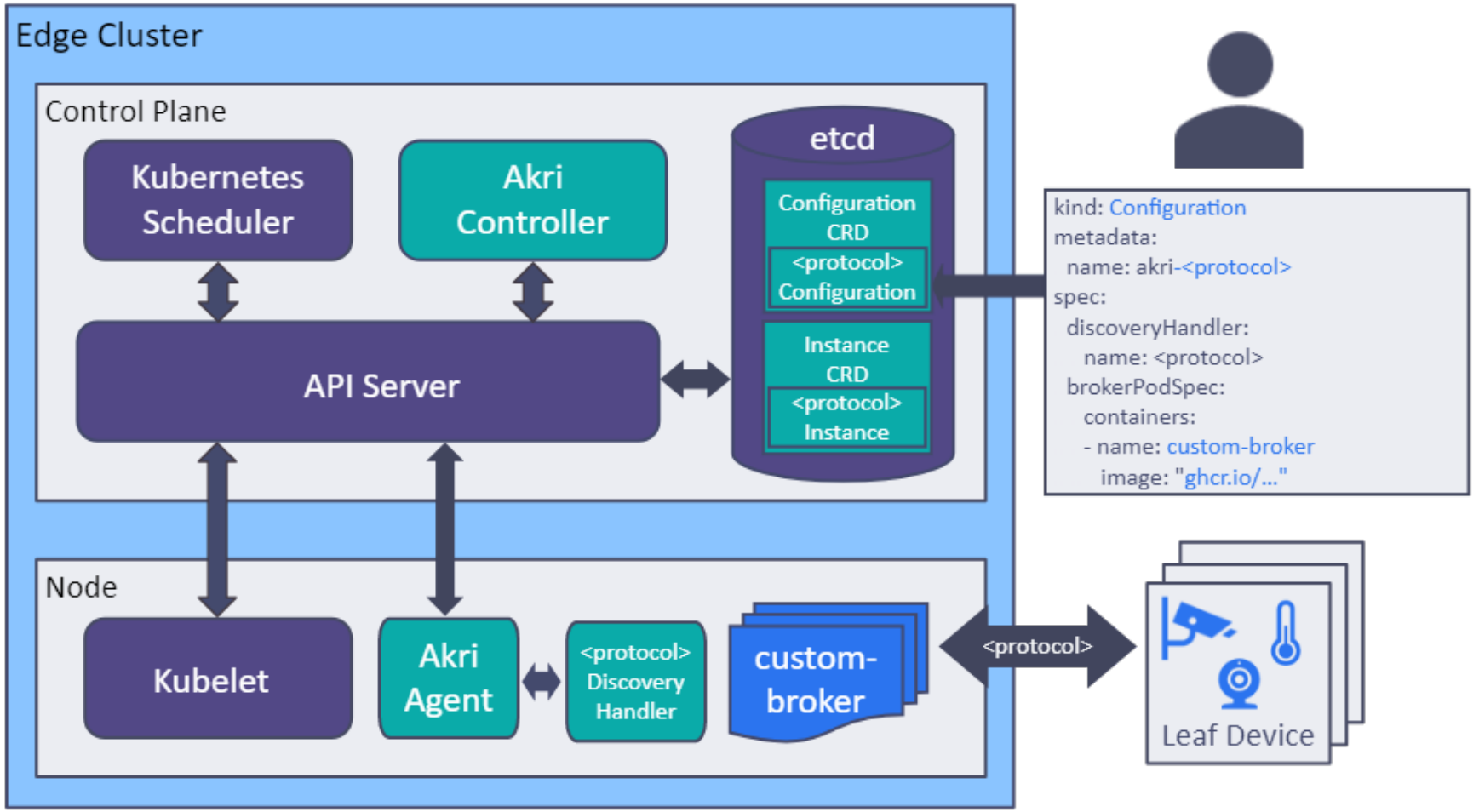
Cloud Native with Edge Infra

Cloud Native with Edge Infra

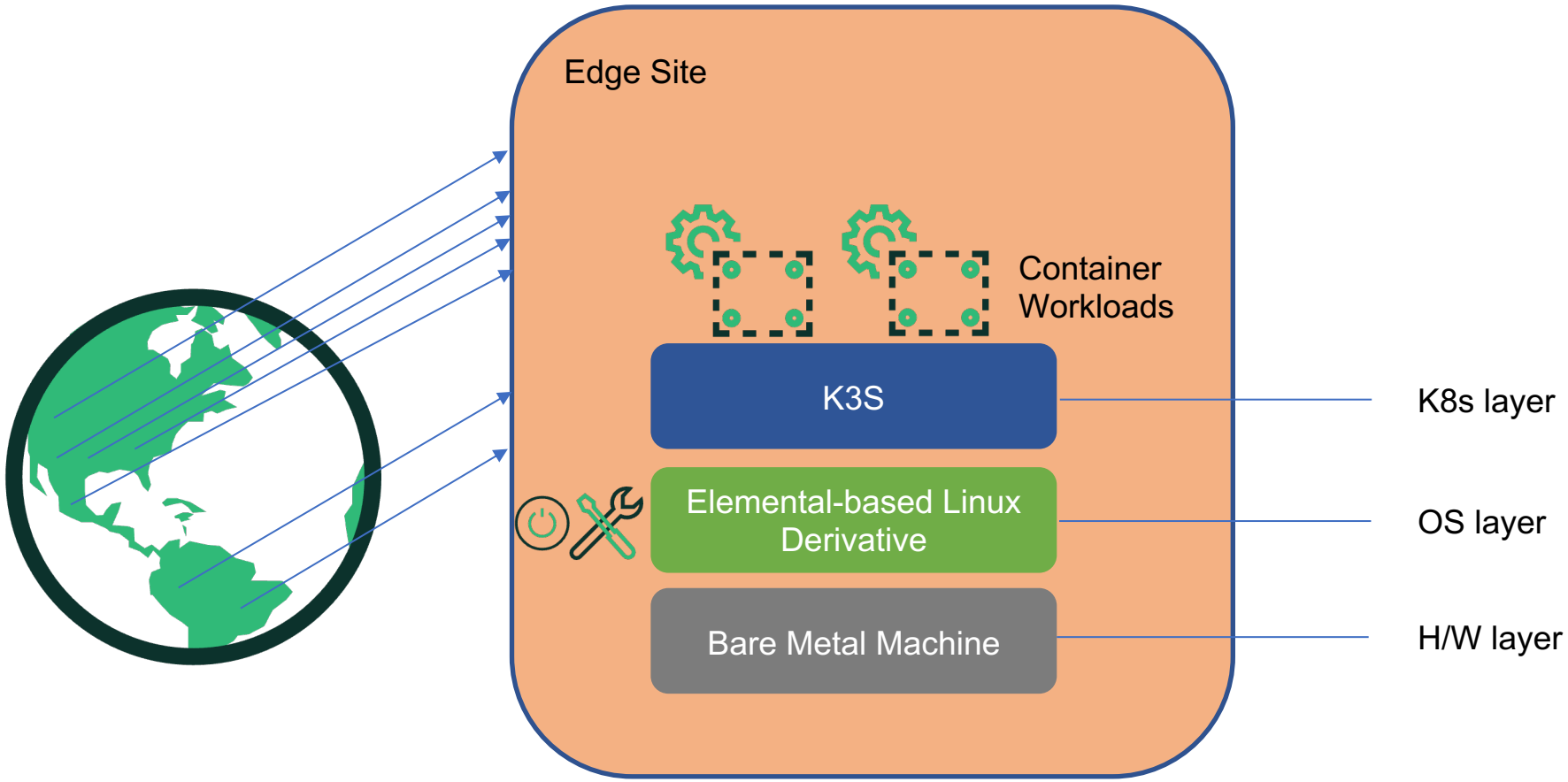
KubeCon CloudNativeCon Europe 2023

Project Akri

CNCF Sandbox project: https://github.com/project-akri/akri



Solving the Edge Puzzle



KubeCon CloudNativeCon
Europe 2023
North America 2021

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