A CI/CD Platform in the Palm of Your Hand

•••

Claudia Beresford

Callisto13

TOC

1. Introduction to MicroVMs
2. What is Liquid Metal?
3. Why is it useful?
4. Demo

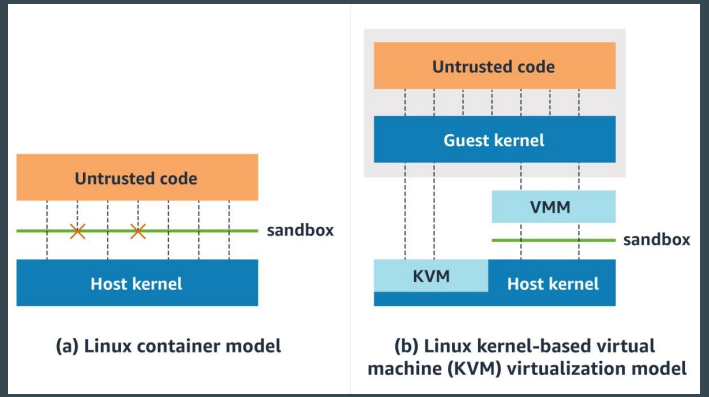
Introduction to MicroVMs

MicroVMs

- Lightweight virtual machine optimising for speed and resource consumption.
- Minimal device emulation support and limited guest functionality resulting in a much smaller footprint.
- Provide a middle ground between containers and VMs
  - Start times of <125ms into userspace
  - Reduced attack surface.
  - Strong process isolation of VMs

Cloud Hypervisor

Firecracker



Benefits

- MicroVMs provide the isolation of VMs without the increased overhead in resource consumption of the host system.
    - Minimal device model excludes non-essential functionality → reduced attack surface.
- Fast, and scalable.
    - <125ms boot & 5 MiB startup RAM
    - Rapid deployment

The Liquid Metal Project?

Liquid Metal

Set of tools to declaratively provision Kubernetes clusters on lightweight VMs (i.e. MicroVMs).

Maintained by Weaveworks.

Comprised of:

- Flintlock
- CAPMVM
- Firecracker / Cloud Hypervisor
- Containerd

Liquidmetal

Flintlock

- Creates and manages the lifecycle of MicroVMs on a group of hosts (bare-metal or virtual).
    - gRPC service, written in Go
- Designed to create MicroVMs on bare-metal hosts.
    - Used as nodes in a virtualised Kubernetes cluster.
- Can be used independently of Liquid Metal.
- Open Source: github.com/weaveworks-liquidmetal/flintlock

Cluster API Provider MicroVM

- ClusterAPI Provider for provisioning Kubernetes clusters on MicroVM nodes.
- Handles the placement of MicroVMs/nodes across a given list of hosts.
- Open Source: github.com/weaveworks-liquidmetal/cluster-api-provider-microvm

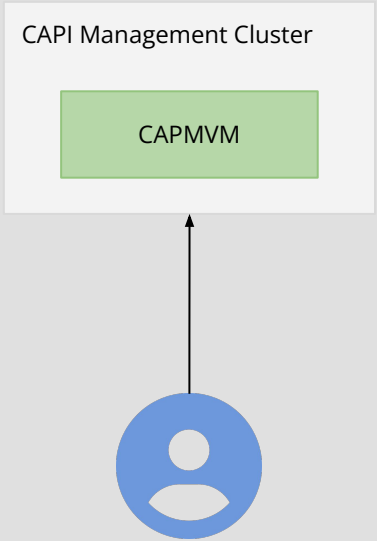
Firecracker & Cloud Hypervisor

- Open source VMMs (Virtual Machine Monitor).
- Firecracker was created by AWS, and is the technology that underpins Lambda and Fargate. (github.com/firecracker-microvm/firecracker)
- Cloud Hypervisor was originally developed by Intel, now under the Linux Foundation. (github.com/cloud-hypervisor/cloud-hypervisor)
- Built on rust-vmm.
- Both projects run on the Linux Kernel-based Virtual Machine (KVM), and exclude unnecessary devices and guest functionality to maintain a very small footprint.

Containerd

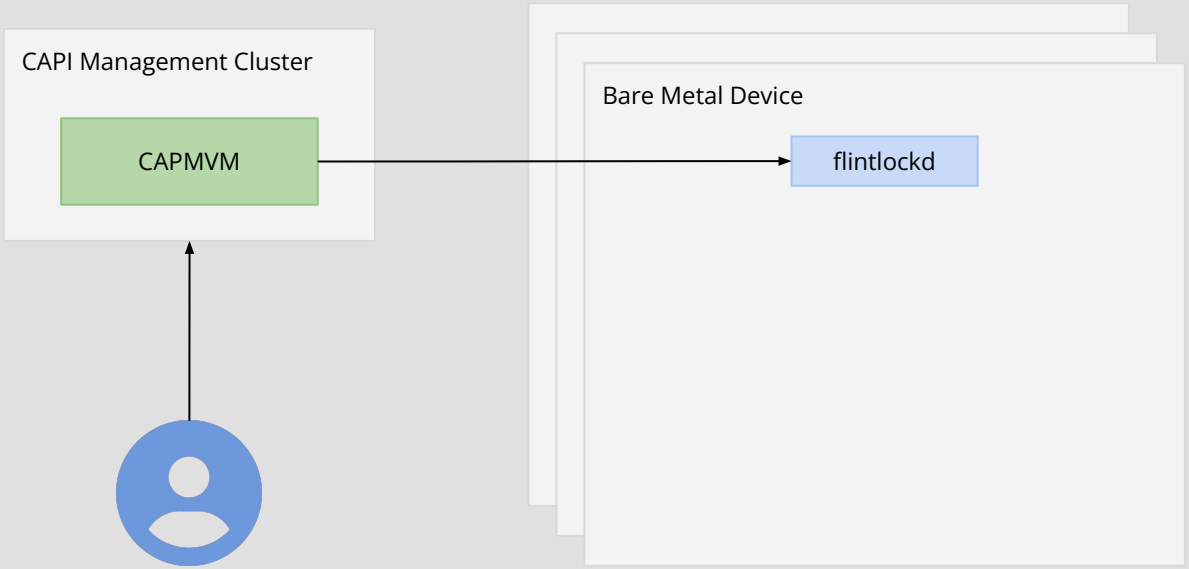
- Daemon that manages the complete container lifecycle of its host system.
    - Image transfer and storage.
    - Container execution and supervision.
    - Network and storage attachments.
- Within Liquid Metal:
    - containerd pulls and manage images which serve as the Kernel and OS volume mounts for each MicroVM.

Architecture



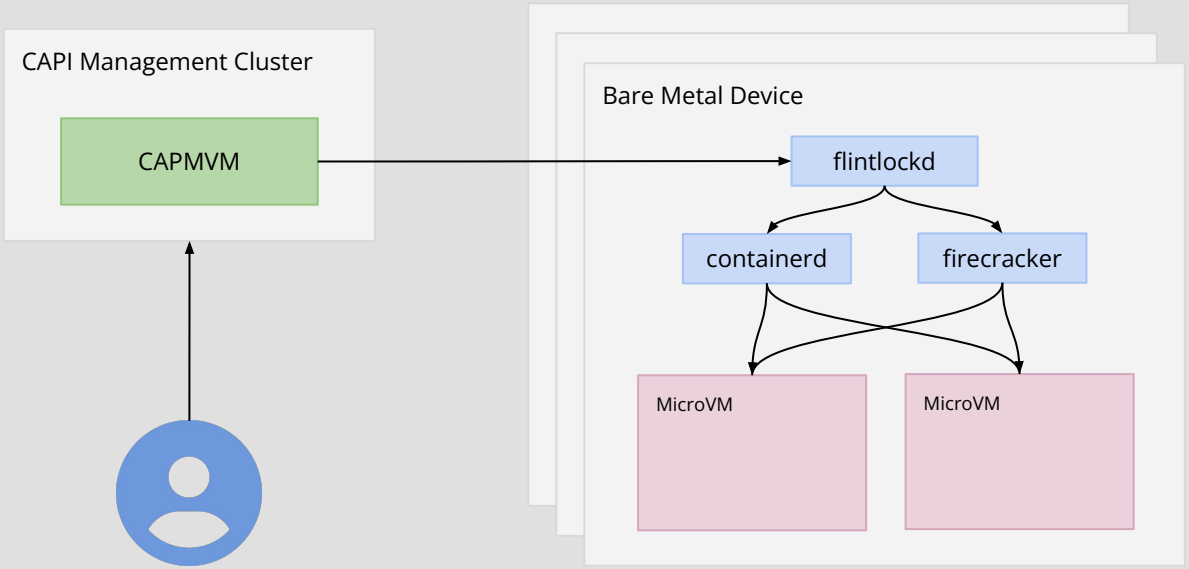
1. Operator applies CAPMVM manifest to CAPI mgmt cluster to create bare metal workloads.

Architecture



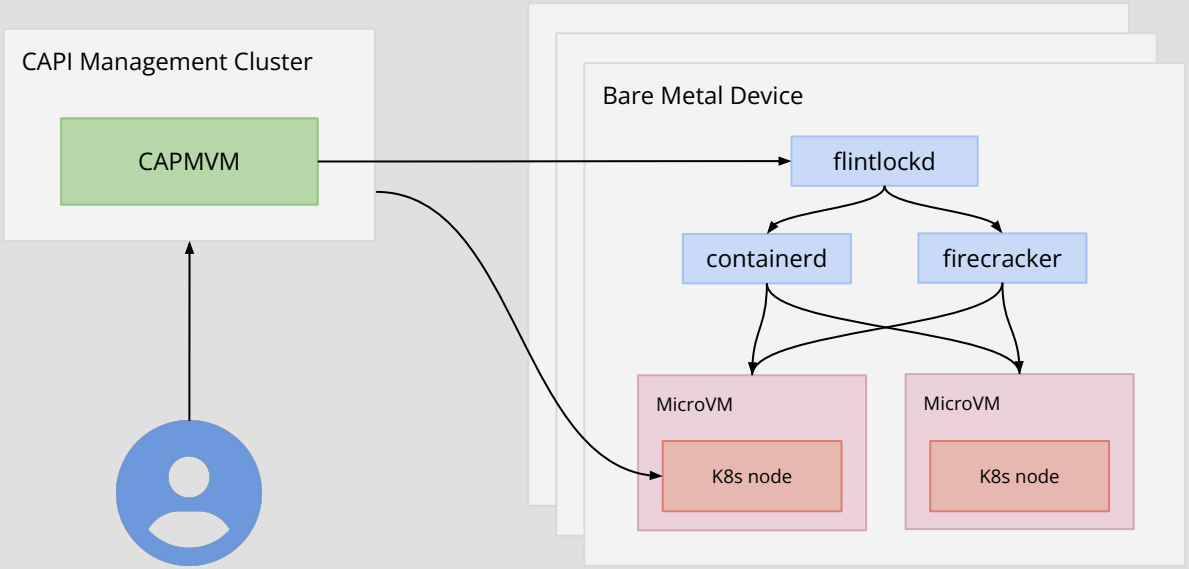
1. Operator applies CAPMVM manifest to CAPI mgmt cluster to create bare metal workloads.
2. CAPMVM creates required nodes via flintlockd running on hosts.

Architecture



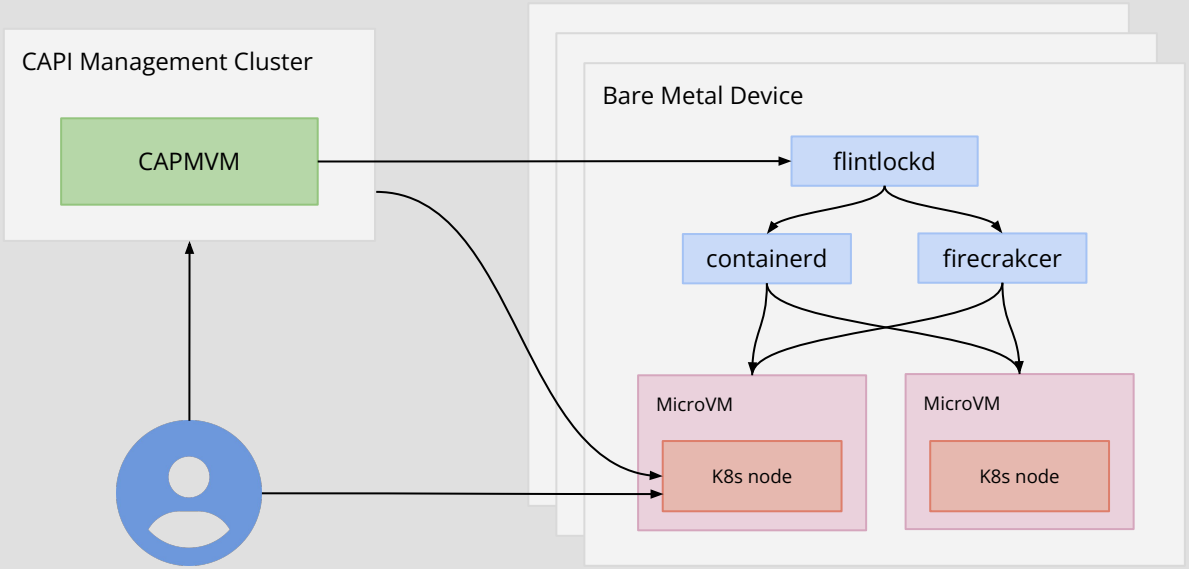
1. Operator applies CAPMVM manifest to CAPI mgmt cluster to create bare metal workloads.
2. CAPMVM creates required nodes via flintlockd running on hosts.
3. Flintlock delegates to containerd to pull kernel & OS images, then starts MicroVM processes.

Architecture



1. Operator applies CAPMVM manifest to CAPI mgmt cluster to create bare metal workloads.
2. CAPMVM creates required nodes via flintlockd running on hosts.
3. Flintlock delegates to containerd to pull kernel & OS images, then starts MicroVM processes.
4. CAPI providers start kubelets on MicroVM hosts, registering them to cluster.

Architecture



1. Operator applies CAPMVM manifest to CAPI mgmt cluster to create bare metal workloads.
2. CAPMVM creates required nodes via flintlockd running on hosts.
3. Flintlock delegates to containerd to pull kernel & OS images, then starts MicroVM processes.
4. CAPI providers start kubelets on MicroVM hosts, registering them to cluster.
5. Operator uses kubeconfig returned by CAPI mgmt cluster to query new workload cluster.

Use cases

Use cases

1. Edge Computing
2. Low resource systems (like homelabs)
3. Bare metal
4. CI self-hosted runners

Demo!!!

Use case: CI/CD Platform

- Proof of concept CI/CD platform
- Using self-hosted Github Actions
- Liquid Metal cluster
  - On bare metal
  - Mix of pod and dedicated MVM runners
    - https://github.com/actions/actions-runner-controller
    - https://github.com/weaveworks-liquidmetal/microvm-action-runner

Actions Runner Controller

- Open Source project
- Kube controller
- Lets you schedule pools of runners as pods
  - Ephemeral
- github.com/github-actions/actions-runner-controller

MicroVM Action Runner

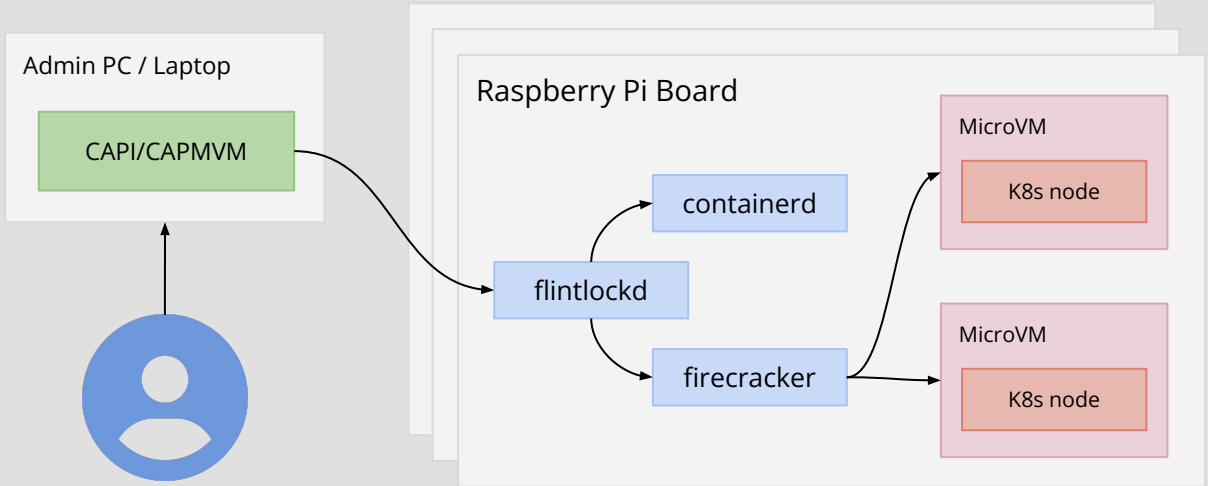
- POC (cannot stress this enough!)
- HTTP service, responds to Github webhook
- Creates ad-hoc MicroVMs to run one-off jobs in full isolation
    - In future hope to add scalable pools
- github.com/weaveworks-liquidmetal/microvm-action-runner

Benefits

- More performant
  - Higher utilisation of runner infra
- More security
  - Flexibility of container builds but with own kernel
  - Ephemeral MVM runners: no risk of cross-pollution from shared builds
- Can run tests in kernel space, eg eBPF and Liquid Metal itself

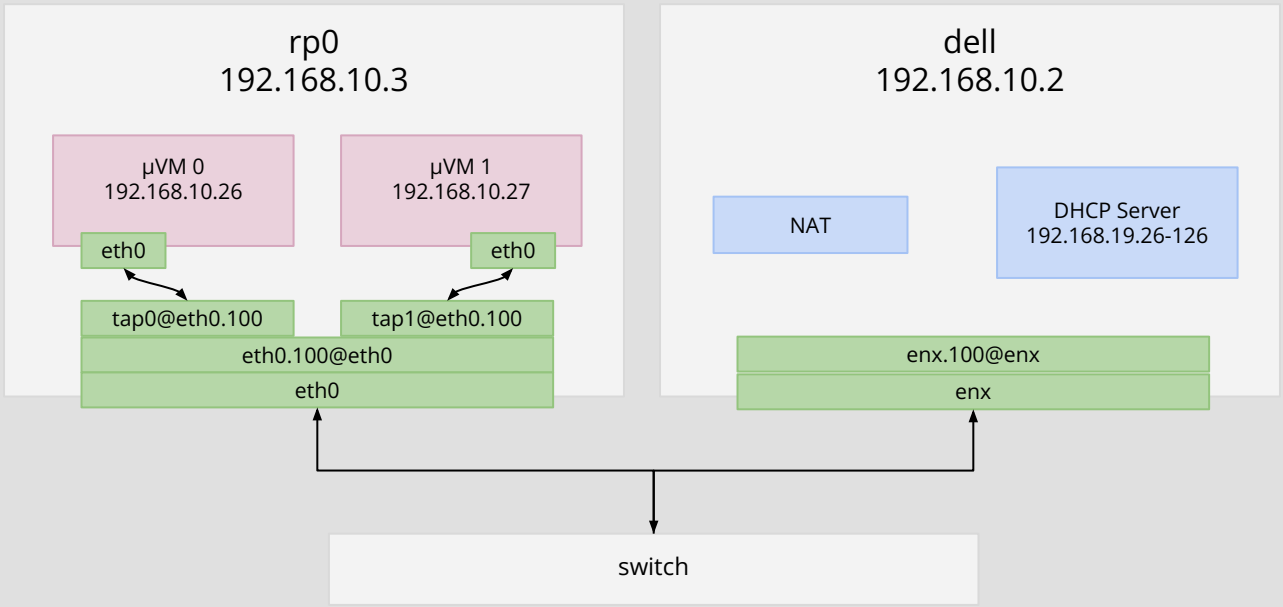
- Environmental benefits:
  - Faster environment builds reduce time to result
  - Lower resource envs lower overall cost of compute
  - MVM images reduce disk usage by sharing kernel/OS layers across builds
  - Underutilized hardware can be pulled into service

The Setup

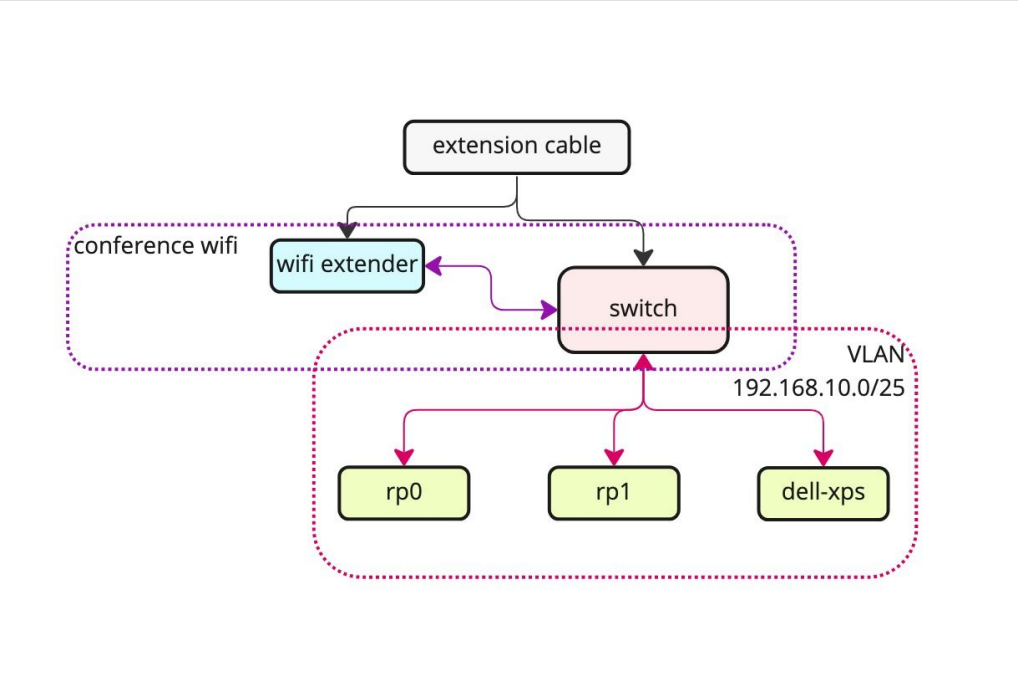


1. Apply CAPMVM cluster manifest to kind cluster.
2. CAPMVM controller asks flintlock service on each board to create required MicroVMs
3. Flintlock delegates to containerd to pull kernel & OS images, then starts MicroVM processes via Firecracker.
4. Kubelets are started on MicroVMs.
5. Kubconfig is returned by mgmt cluster to query new homelab cluster

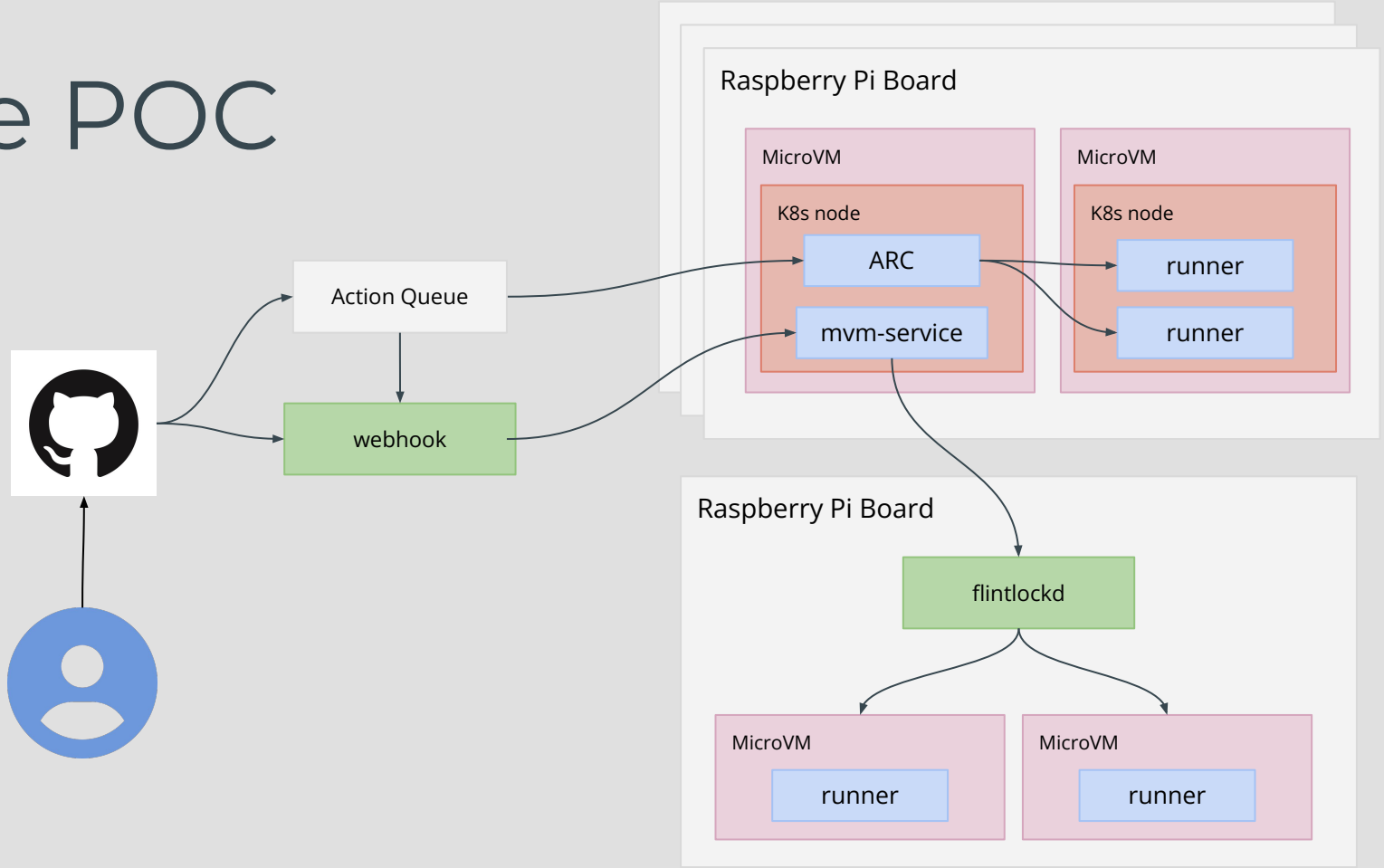
The Network


rp0
192.168.10.3
µVM 0
192.168.10.26
µVM 1
192.168.10.27
eth0
eth0
tap0@eth0.100
tap1@eth0.100
eth0.100@eth0
eth0
dell
192.168.10.2
NAT
DHCP Server
192.168.19.26-126
enx.100@enx
enx
switch

The Hackery


extension cable
conference wifi
wifi extender
switch
VLAN
192.168.10.0/25
rp0
rp1
dell-xps

The POC

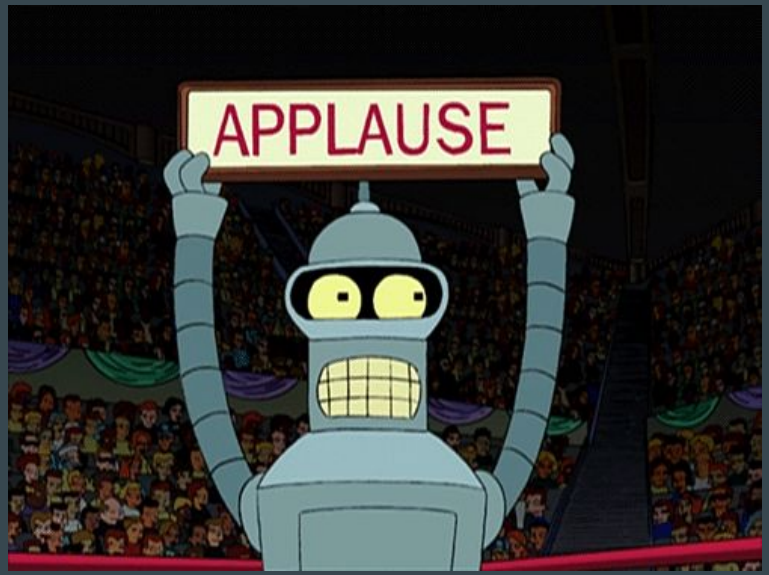

Raspberry Pi Board
MicroVM
K8s node
ARC
mvm-service
MicroVM
K8s node
runner
runner
Action Queue
webhook
Raspberry Pi Board
flintlockd
MicroVM
runner
MicroVM
runner


LET'S GO, ALREADY!

Learnings

- Github is not in the least reliable
  - Known issue of jobs not being picked up by idle runners, sometimes waiting up to 20 minutes!
- *Theoretically* more cost efficient than a standard Enterprise Github Action package

Thanks for watching!



Docs

HomeLab Docs:
bit.ly/cosmic-homelab

Liquid Metal Docs:
bit.ly/ww-liquid-metal