Rust China Meetup
Hangzhou 2021

Cloud-Hypervisor on ARM

Feng Li (李枫)
hkli2013@126.com
Aug 8，2021

Agenda

I. Background
- Virtualization
- Rust-VMM
- Cloud-Hypervisor
- Kata Containers
- Testbeds

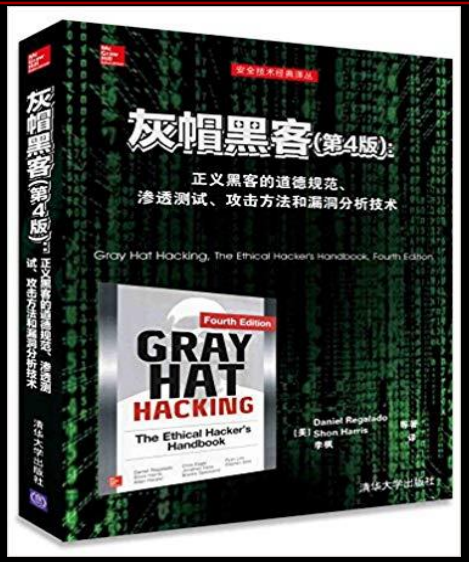
II. Cloud-Hypervisor + Kata Containers
- Cloud-Hypervisor on RPi4
- Kata Containers on RPi4

III. Rust for Cloud Native
- Rust for K8S
- Good Resources

IV. Wrap-up

Who Am I

- The main translator of the book «Gray Hat Hacking The Ethical Hacker's Handbook, Fourth Edition» (ISBN：9787302428671) & «Linux Hardening in Hostile Networks, First Edition» (ISBN: 9787115544384)



- Pure software developmement for ~15 years
- Actively participate in various activities of the open source community
- https://github.com/XianBeiTuoBaFeng2015/MySlides/tree/master/Conf
  https://github.com/XianBeiTuoBaFeng2015/MySlides/tree/master/LTS
- Recently, focus on infrastructure of Cloud/Edge Computing, AI, Virtualization, Program Runtimes, Network, 5G, RISC-V, EDA…

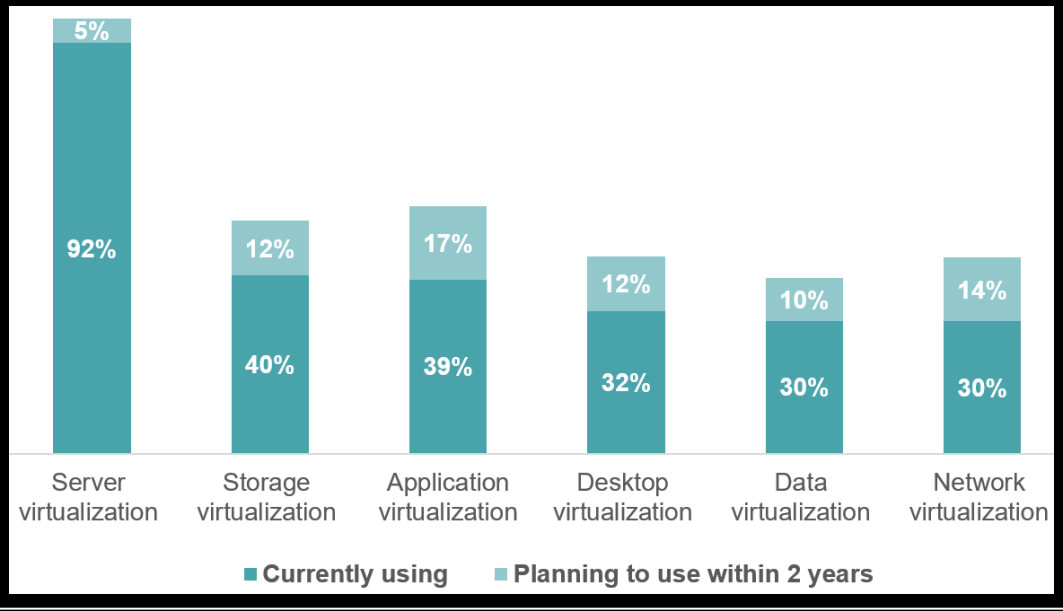
I. Overview

1) Virtualization

- https://en.wikipedia.org/wiki/Virtualization

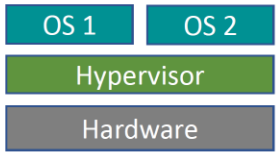
In computing, virtualization or virtualisation (sometimes abbreviated v12n, a numeronym) is the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources.

- https://www.spiceworks.com/marketing/reports/state-of-virtualization/

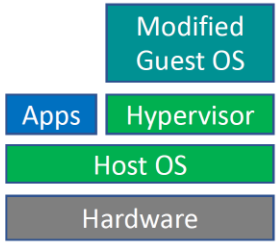


Virtualization Types

- Fundamental types
  - **Type 1: Full-virtualization**, where the hypervisor takes control of the hardware and hosts the guest OSes, and the guests are completely unaware of running on an virtualized environment.
  - **Type 2: Para-virtualization**, where one of the operating system (called as Host OS) takes charge of hardware and the guest OS is modified to connect with either Host OS or hardware devices.
- Derived Types
  - **Hardware assisted virtualization**: Here the virtualization solution utilizes the support provided by hardware to realize the virtualization goals.
    - Example Linux/KVM falls under this category. We will see this in details, later.
  - **Hybrid types**: Here the virtualization is realized by combining different other types.
    - For example, the core virtualization functions are realized using Type 1 hypervisor and peripheral / device virtualization are done using Type 2 or other types such as Graphics, Display virtualization uses a server in Host OS and clients running in Guests OSes. We will see this in details, later.
  - ... and many more



Type 1



Type 2

Source: https://www.slideshare.net/AananthCN/virtualization-support-in-armv8

System Virtualization involves following functions:

- Virtualization of CPU cores or Processing Elements
- Virtualization of memory and the memory management
- Virtualization of Interrupts
- Virtualization of Timers
- I/O or Peripheral Virtualization

...

Source: https://www.slideshare.net/AananthCN/virtualization-support-in-armv8

- **SW**

Comparison of platform virtualization software			
Hardware (hypervisors)	Native		Adeos • CP/CMS • Hyper-V • KVM (oVirt • Red Hat Enterprise Virtualization) • LDoms / Oracle VM Server for SPARC • Logical Partition (LPAR) • LynxSecure • PikeOS • Proxmox VE • QNX (QNX Hypervisor • QNX Hypervisor for Safety) • SIMMON • VMware ESXi (VMware vSphere • vCloud) • VMware Infrastructure • Xen (Oracle VM Server for x86 • XenServer) • XtratuM • z/VM
	Hosted	Specialized	Basilisk II • Bochs • Cooperative Linux • DOSBox • DOSBox-X • DOSEMU • PCem • PikeOS • SheepShaver • SIMH • Windows on Windows (Virtual DOS machine) • Win4Lin
		Independent	bhyve • Microsoft Virtual Server • Parallels Workstation • Parallels Desktop for Mac • Parallels Server for Mac • PearPC • QEMU • VirtualBox • Virtual Iron • VMware Fusion • VMware Server • VMware Workstation (Player) • Windows Virtual PC
	Tools		Ganeti • System Center Virtual Machine Manager • Virtual Machine Manager
Operating system	OS containers		FreeBSD jail • iCore Virtual Accounts • Linux-VServer • LXC • OpenVZ • Solaris Containers • Virtuozzo • Workload Partitions
	Application containers		Docker • lmctfy • rkt
	Virtual kernel architectures		Rump kernel • User-mode Linux • vkernel
	Related kernel features		BrandZ • cgroups • chroot • namespaces • seccomp
	Orchestration		Amazon ECS • Kubernetes • OpenShift
Desktop			Citrix XenApp • Citrix XenDesktop • Remote Desktop Services • VMware Horizon View • Ulteo Open Virtual Desktop
Application			Ceedo • Citrix XenApp • Dalvik • InstallFree • Microsoft App-V • Remote Desktop Services • Symantec Workspace Virtualization • Turbo • VMware ThinApp • ZeroVM
Network			Distributed Overlay Virtual Ethernet (DOVE) • Ethernet VPN (EVPN) • NVGRE • Open vSwitch • Virtual security switch • Virtual Extensible LAN (VXLAN)
See also: List of emulators			

https://en.wikipedia.org/wiki/Comparison_of_platform_virtualization_software

- **https://en.wikipedia.org/wiki/Timeline_of_virtualization_development**
- **https://en.wikipedia.org/wiki/Emulator**

- **https://en.wikipedia.org/wiki/OS-level_virtualization**
- **https://en.wikipedia.org/wiki/List_of_Linux_containers**
- **https://en.wikipedia.org/wiki/Open_Container_Initiative**
- **https://en.wikipedia.org/wiki/Docker_(software)**

1.1 Hypervisor or VMM(Virtual Machine Monitor)

Overview

A **hypervisor** (or **virtual machine monitor**, **VMM**, **virtualizer**) is a kind of emulator; it is computer software, firmware or hardware that creates and runs virtual machines. A computer on which a hypervisor runs one or more virtual machines is called a *host machine*, and each virtual machine is called a *guest machine*. The hypervisor presents the guest operating systems with a virtual operating platform and manages the execution of the guest operating systems. Multiple instances of a variety of operating systems may share the virtualized hardware resources: for example, Linux, Windows, and macOS instances can all run on a single physical x86 machine. This contrasts with operating-system-level virtualization, where all instances (usually called *containers*) must share a single kernel, though the guest operating systems can differ in user space, such as different Linux distributions with the same kernel.

The term *hypervisor* is a variant of *supervisor*, a traditional term for the kernel of an operating system: the hypervisor is the supervisor of the supervisors,[1] with *hyper-* used as a stronger variant of *super-*.[a] The term dates to circa 1970;[2] in the earlier CP/CMS (1967) system, the term *Control Program* was used instead.

Classification [edit]

In their 1974 article, *Formal Requirements for Virtualizable Third Generation Architectures*, Gerald J. Popek and Robert P. Goldberg classified two types of hypervisor:[3]

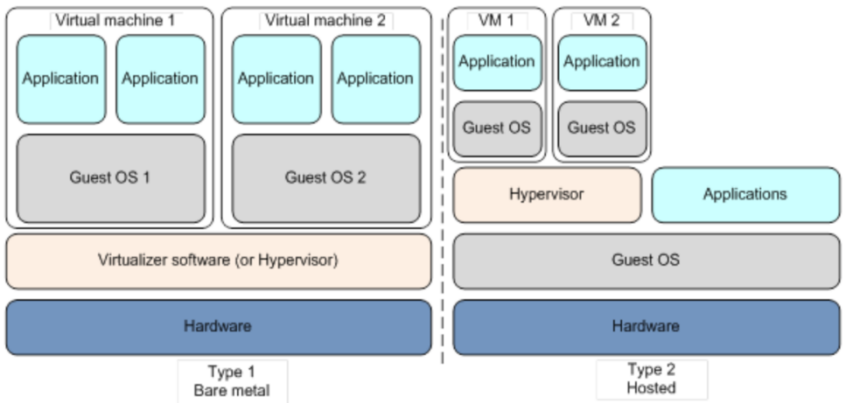
Type-1, native or bare-metal hypervisors

These hypervisors run directly on the host's hardware to control the hardware and to manage guest operating systems. For this reason, they are sometimes called bare metal hypervisors. The first hypervisors, which IBM developed in the 1960s, were native hypervisors.[4] These included the test software SIMMON and the CP/CMS operating system (the predecessor of IBM's z/VM). Modern equivalents include AntsleOS,[5] Microsoft Hyper-V and Xbox One system software, Nutanix AHV, XCP-ng, Oracle VM Server for SPARC, Oracle VM Server for x86, POWER Hypervisor,[6] QNX Hypervisor,[7] VMware ESXi (formerly ESX), Proxmox Virtual Environment and Xen.

Type-2 or hosted hypervisors

These hypervisors run on a conventional operating system (OS) just as other computer programs do. A guest operating system runs as a process on the host. Type-2 hypervisors abstract guest operating systems from the host operating system. Parallels Desktop for Mac, QEMU, VirtualBox, VMware Player and VMware Workstation are examples of type-2 hypervisors.

The distinction between these two types is not always clear. For instance, Linux's Kernel-based Virtual Machine (KVM) and FreeBSD's bhyve are kernel modules[8] that effectively convert the host operating system to a type-1 hypervisor.[9] At the same time, since Linux distributions and FreeBSD are still general-purpose operating systems, with applications competing with each other for VM resources, KVM and bhyve can also be categorized as type-2 hypervisors.[10]



https://en.wikipedia.org/wiki/Virtual_machine

KVM (Kernel-based Virtual Machine)

- https://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine

Kernel-based Virtual Machine (**KVM**) is a virtualization module in the Linux kernel that allows the kernel to function as a hypervisor. It was merged into the mainline Linux kernel in version 2.6.20, which was released on February 5, 2007.[1] KVM requires a processor with hardware virtualization extensions, such as Intel VT or AMD-V.[2] KVM has also been ported to other operating systems such as FreeBSD[3] and illumos[4] in the form of loadable kernel modules.

KVM was originally designed for x86 processors but has since been ported to S/390,[5] PowerPC,[6] IA-64, and ARM.[7]

KVM provides hardware-assisted virtualization for a wide variety of guest operating systems including Linux, BSD, Solaris, Windows, Haiku, ReactOS, Plan 9, AROS Research Operating System and macOS.[8][9] In addition, Android 2.2, GNU/Hurd[10] (Debian K16), Minix 3.1.2a, Solaris 10 U3 and Darwin 8.0.1, together with other operating systems and some newer versions of these listed, are known to work with certain limitations.[11]

Additionally, KVM provides paravirtualization support for Linux, OpenBSD,[12] FreeBSD,[13] NetBSD,[14] Plan 9[15] and Windows guests using the VirtIO API.[16] This includes a paravirtual Ethernet card, disk I/O controller,[17] balloon driver, and a VGA graphics interface using SPICE or VMware drivers.

https://www.linux-kvm.org/page/Main_Page
https://git.kernel.org/pub/scm/virt/kvm/kvm.git
https://en.wikipedia.org/wiki/Qumranet

- Internals:

KVM provides device abstraction but no processor emulation. It exposes the `/dev/kvm` interface, which a user mode host can then use to:

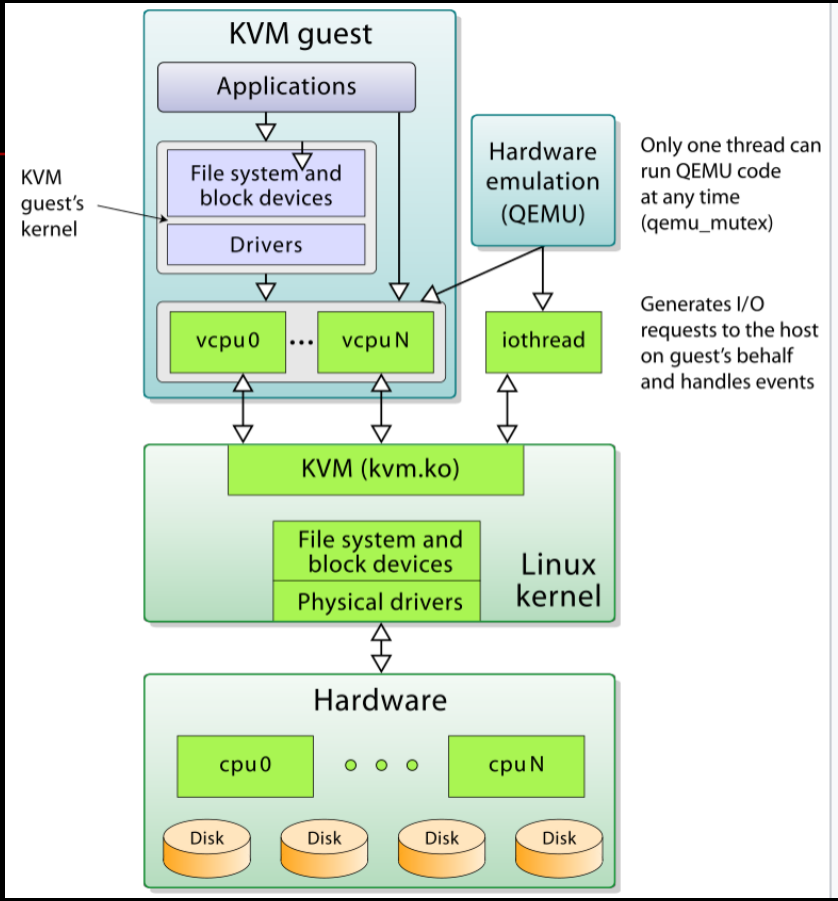
- Set up the guest VM's address space. The host must also supply a firmware image (usually a custom BIOS when emulating PCs) that the guest can use to bootstrap into its main OS.
- Feed the guest simulated I/O.
- Map the guest's video display back onto the system host.

On Linux, QEMU versions 0.10.1 and later is one such userspace host. QEMU uses KVM when available to virtualize guests at near-native speeds, but otherwise falls back to software-only emulation.

Internally, KVM uses SeaBIOS as an open source implementation of a 16-bit x86 BIOS.[23]

-

- **A high-level overview of the KVM/QEMU virtualization environment:**



QEMU

- https://en.wikipedia.org/wiki/QEMU
  https://www.qemu.org/

QEMU (short for **Q**uick **EMU**lator[2][*dubious – discuss*]) is a free and open-source emulator and virtualizer that can perform hardware virtualization.

QEMU is a hosted virtual machine monitor: it emulates the machine's processor through dynamic binary translation and provides a set of different hardware and device models for the machine, enabling it to run a variety of guest operating systems. It also can be used with Kernel-based Virtual Machine (KVM) to run virtual machines at near-native speed (by taking advantage of hardware extensions such as Intel VT-x). QEMU can also do emulation for user-level processes, allowing applications compiled for one architecture to run on another.[3]

- Operating modes:

User-mode emulation
In this mode QEMU runs single Linux or Darwin/macOS programs that were compiled for a different instruction set. System calls are thunked for endianness and for 32/64 bit mismatches. Fast cross-compilation and cross-debugging are the main targets for user-mode emulation.

System emulation
In this mode QEMU emulates a full computer system, including peripherals. It can be used to provide virtual hosting of several virtual computers on a single computer. QEMU can boot many guest operating systems, including Linux, Solaris, Microsoft Windows, DOS, and BSD;[6] it supports emulating several instruction sets, including x86, MIPS, 32-bit ARMv7, ARMv8, PowerPC, SPARC, ETRAX CRIS and MicroBlaze.

KVM Hosting
Here QEMU deals with the setting up and migration of KVM images. It is still involved in the emulation of hardware, but the execution of the guest is done by KVM as requested by QEMU.

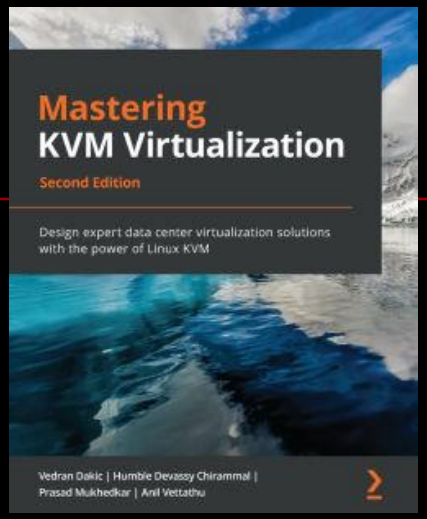
Xen Hosting
QEMU is involved only in the emulation of hardware; the execution of the guest is done within Xen and is totally hidden from QEMU.

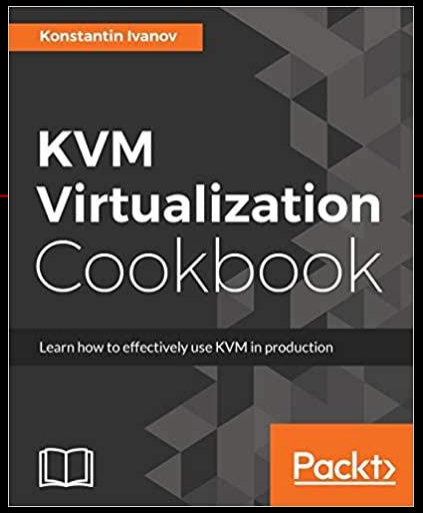
- TCG(Tiny Code Generator):

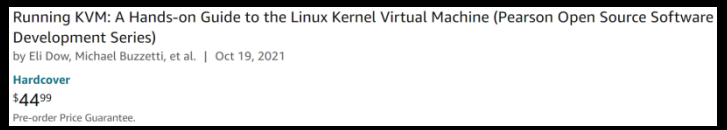
The Tiny Code Generator (TCG) aims to remove the shortcoming of relying on a particular version of GCC or any compiler, instead incorporating the compiler (code generator) into other tasks performed by QEMU at run time. The whole translation task thus consists of two parts: basic blocks of target code (*TBs*) being rewritten in *TCG ops* - a kind of machine-independent intermediate notation, and subsequently this notation being compiled for the host's architecture by TCG. Optional optimisation passes are performed between them, for a just-in-time compiler (JIT) mode.

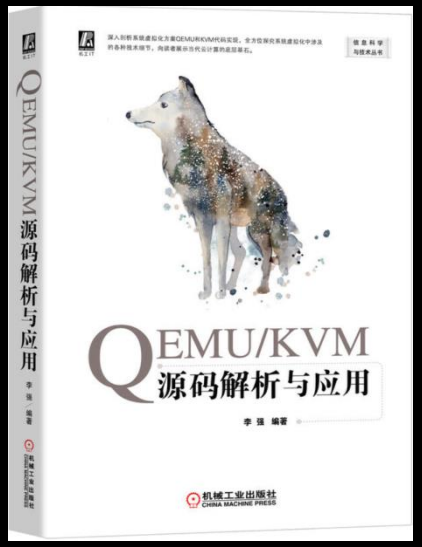
TCG requires dedicated code written to support every architecture it runs on, so that the JIT knows what to translate the *TCG ops* to. If no dedicated JIT code is available for the architecture, TCG falls back to a slow interpreter mode called TCG Interpreter (TCI). It also requires updating the target code to use TCG ops instead of the old *dyngen* ops.

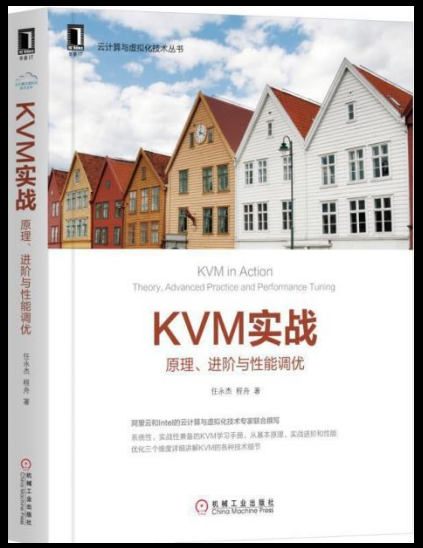
Good Resource

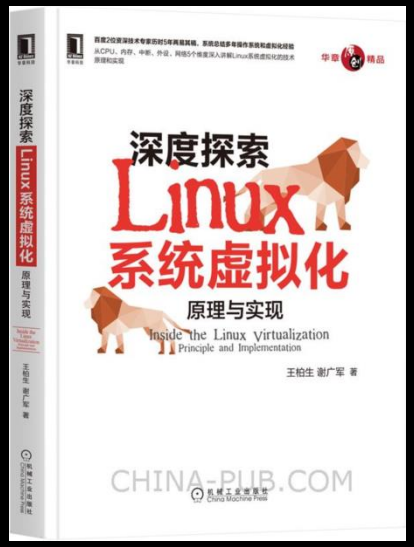












- https://events.linuxfoundation.org/kvm-forum/

1.2 Hardware-assisted

Overview

- https://en.wikipedia.org/wiki/Hardware-assisted_virtualization

In computing, **hardware-assisted virtualization** is a platform virtualization approach that enables efficient full virtualization using help from hardware capabilities, primarily from the host processors. A full virtualization is used to emulate a complete hardware environment, or virtual machine, in which an unmodified guest operating system (using the same instruction set as the host machine) effectively executes in complete isolation. Hardware-assisted virtualization was added to x86 processors (Intel VT-x or AMD-V) in 2005 and 2006 (respectively).

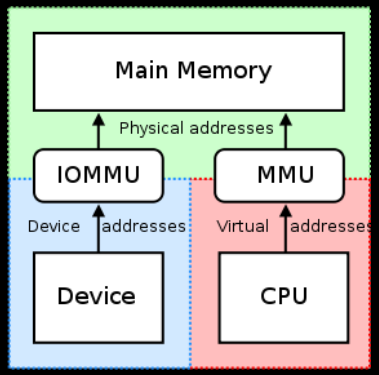
Hardware-assisted virtualization is also known as **accelerated virtualization**; Xen calls it **hardware virtual machine** (**HVM**), and Virtual Iron calls it **native virtualization**.

e.g., https://en.wikipedia.org/wiki/X86_virtualization

https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/virtualization-enabling-intel-virtualization-technology-features-and-benefits-paper.pdf

- IOMMU

https://en.wikipedia.org/wiki/Input-output_memory_management_unit



- ...

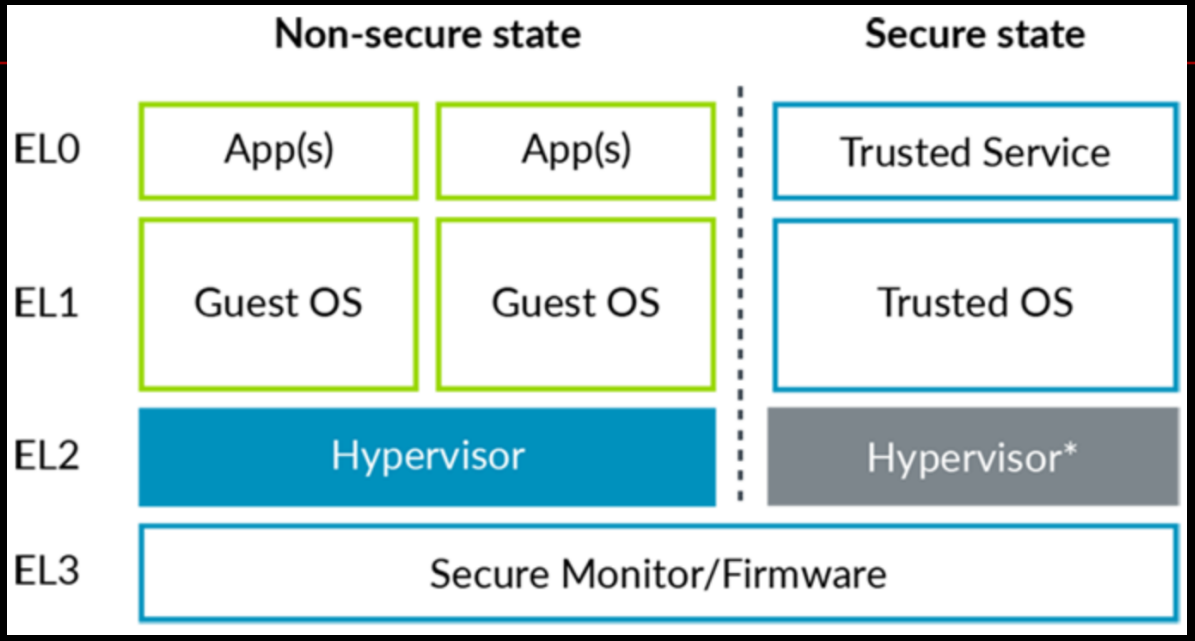
Good Resources

- http://cdn.ttgtmedia.com/searchServerVirtualization/downloads/essential+guide+to+choosing+virtualization+hardware.pdf
- https://act-on.ioactive.com/acton/attachment/34793/f-4bfc17b4-7b5d-4f37-a6f1-d1cb863f08c1/1/-/-/-/-/Intel-AMD-Cross-Platform-comp.pdf
- https://lenovopress.com/lp1467.pdf
- …

1.3 ARM

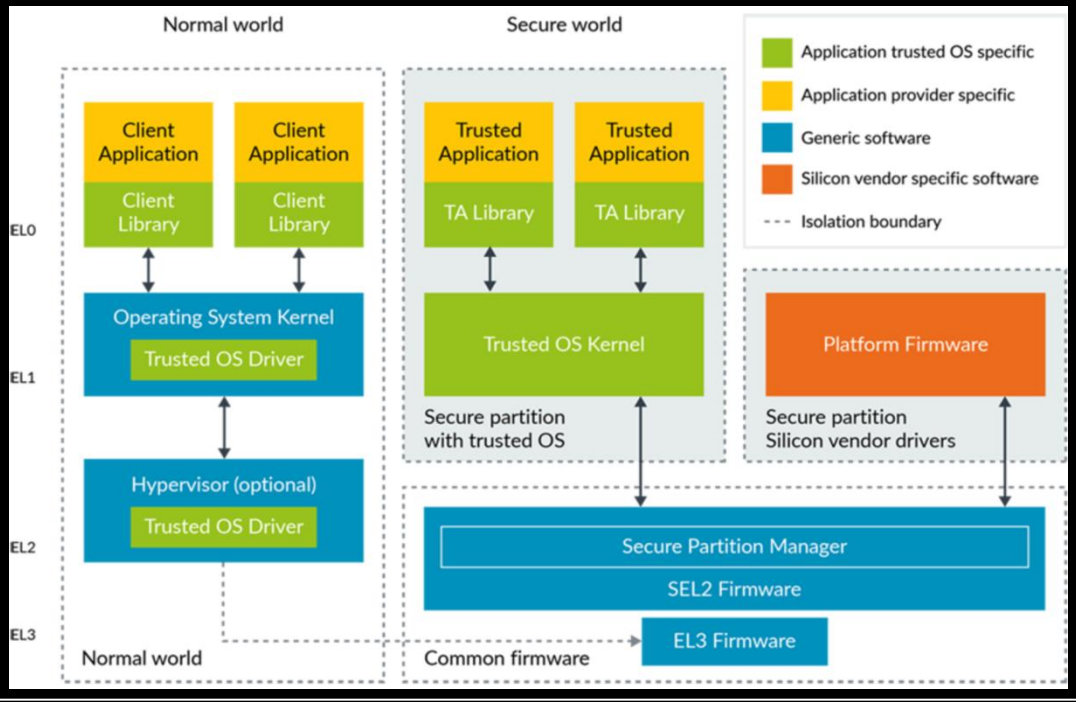
Cortex-A

- https://developer.arm.com/documentation/100942/0100/



- https://developer.arm.com/documentation/102142/0100/Virtualizating-exceptions
- https://developer.arm.com/architectures/cpu-architecture/a-profile?_ga=2.206541139.1550823940.1628221907-1061672970.1627236364
- https://developer.arm.com/documentation/102142/0100/Introduction-to-virtualization

- **Virtualization Host Extensions**
  https://developer.arm.com/documentation/102142/0100/Virtualization-Host-Extensions
- **Nested Virtualization Extensions**
  https://developer.arm.com/documentation/102142/0100/Nested-virtualization
- **Secure Virtualization**
  https://developer.arm.com/documentation/102142/0100/Secure-virtualization



- **System MMU(SMMU)**
  https://en.wikipedia.org/wiki/Input-output_memory_management_unit
  https://developer.arm.com/ip-products/system-ip/system-controllers/system-memory-management-unit

Segments	Use cases	Hypervisors	Whitepaper year and related ARM CoreLink System IP
Mobile	BYOD	VMware MVP	2011 - SMMU Automotive
Automotive	ADAS/IVI ECU consolidation	Green Hills (INTEGRITY) VirtualLogix	
Server	Live migration Rapid deployment Sandboxing	Xen / KVM	2017 - SMMU & GIC

- **Generic Interrupt Controller(GIC)**
  https://developer.arm.com/ip-products/system-ip/system- controllers/interrupt-controllers
- **The Generic Timers**
  https://developer.arm.com/documentation/102142/0100/Virtualizating-the-Generic-Timers
- **…**

The evolution of ARM Virtualization

- **ARMv8.0-A**

  EL2 is Not EL1++:

  - EL2 is not a superset of NS-EL1
    - Orthogonal mode to EL1
    - Allows multiplexing of NS-EL1 guests on the hardware
  - Own translation regime
    - Separate Stage-1 translation, no Stage-2 translation
  - It would be difficult to run Linux in EL2
    - Requires too many changes to be practical
  - EL2 could be used as a ”world switch”
    - Between guests (baremetal hypervisor/Type I)
    - Between host and guest (hosted hypervisor/Type II)

    This makes the host a form of specialized guest.

Source: “The evolution of Virtualization in the Arm Architecture” , Julien Grall, XPDDS 2018.

ARMv8.1-A
EL2 enhancement:

The Virtualization Host Extension (VHE) expands the capability of EL2:

- Designed to improve the support of the Type-2 hypervisors
- Allows the host OS to be run at EL2
- The host OS requires minimal changes to run at EL2
- User-space still runs at EL0
- Host has no software running at EL1
- AArch64 specific

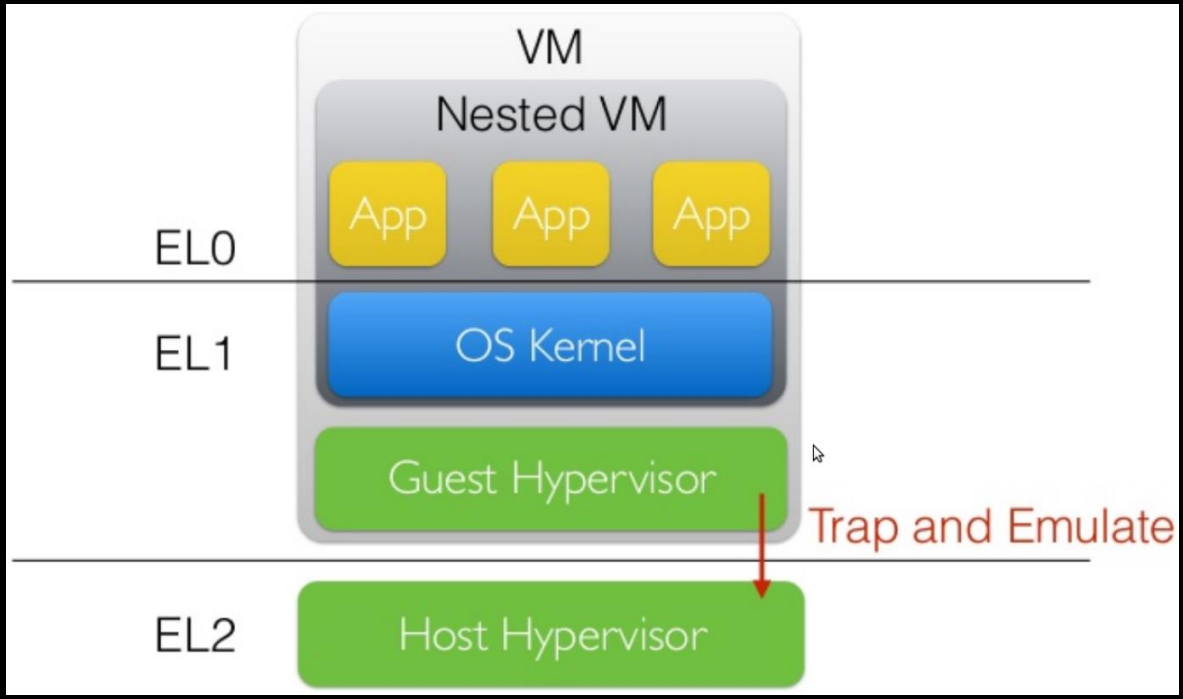
EL2 becomes a strict superset of EL1

Source: "The evolution of Virtualization in the Arm Architecture" , Julien Grall, XPDDS 2018.

ARMv8.3-A Nested Virtualization:

The Nested Virtualization extension allows a hypervisor in a VM.

- Unmodified guest hypervisor running in NS EL1
- Implementation of a host hypervisor required
  - Running at EL2
- AArch64 specific



Source: "The evolution of Virtualization in the Arm Architecture" , Julien Grall, XPDDS 2018.

Memory Partitioning And Monitoring:

The Memory Partitioning And Monitoring (MPAM) allows:

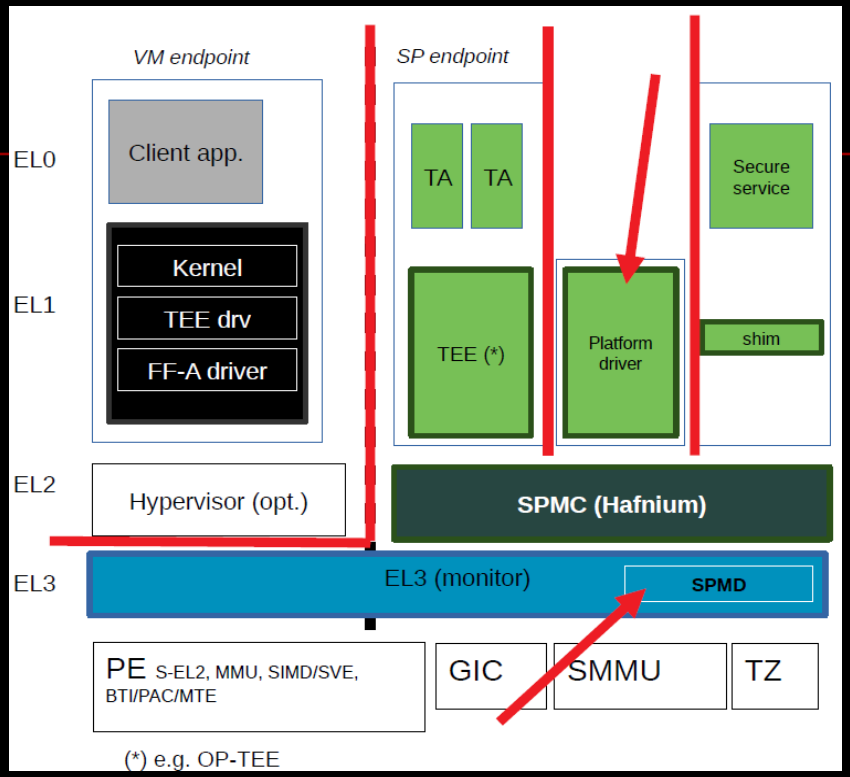
- to limit the memory system performance impact of a VM
  - provide more bandwidth to some processes

A hypervisor can monitor and control how VM uses:

- memory of a system
- communicate with other system components

Source: "The evolution of Virtualization in the Arm Architecture" , Julien Grall, XPDDS 2018.

Armv8.4-A Secure EL2:



Source: "Secure Partition Manager(Armv8.4 Secure EL2)", Olivier Deprez & Madhukar Pappireddy, LVC21-303.

Nested Virtualization:

ARMv8.4 extends Nested Virtualization to:

- Reduce the number of traps
- Improve performance of nested hypervisor

Source: "The evolution of Virtualization in the Arm Architecture" , Julien Grall, XPDDS 2018.

- **Summary**

  ARMv8.1-A: https://goo.gl/Ox4thV

  ARMv8.2-A: https://goo.gl/0Ns37U

  ARMv8.3-A: https://goo.gl/CJv1n0

  ARMv8.4-A: https://goo.gl/tYZmhR

  …

Good Resources

- https://systems.cs.columbia.edu/projects/kvm-arm/
- https://pdfs.semanticscholar.org/cd5c/97e17c352024cdb9e71eeef5480a78a5e7dc.pdf
- https://www.researchgate.net/publication/327327630_ARM_Virtualization
- https://wiki.osdev.org/Generic_Interrupt_Controller
- https://lwn.net/Articles/855972/
- https://www.wikiwand.com/en/AArch64
- https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/Learn%20the%20Architecture/Armv8-A%20virtualization.pdf?revision=a765a7df-1a00-434d-b241-357bfda2dd31
- …

2) Rust-VMM

- https://github.com/rust-vmm/
  Building blocks for VMMs written in Rust
- https://github.com/rust-vmm/community

rust-vmm is an open-source project that empowers the community to build custom Virtual Machine Monitors (VMMs) and hypervisors. It provides a set of virtualization components that any project can use to quickly develop virtualization solutions while focusing on the key differentiators of their product rather than re-implementing common components like KVM wrappers, virtio devices and other VMM libraries.

The rust-vmm project is organized as a shared effort, shared ownership open-source project that includes (so far) contributors from Alibaba, AWS, Cloud Base, Crowdstrike, Intel, Google, Red Hat as well as individual contributors.

Each virtualization component lives in a separate GitHub repository under the rust-vmm organization. One repository corresponds to one Rust crate.

- Why is it

- **Reduce code duplication**. The initial thought with rust-vmm was to create a place for sharing common virtualization components between two existing VMMs written in Rust: CrosVM and Firecracker. These two projects have similar code for calling KVM ioctls, managing the virtual machine memory, interacting with virtio devices and others. Instead of having these components live in each of the projects repositories, they can be shared through the rust-vmm project.
- **Faster development**. rust-vmm provides a base of virtualization components which are meant to be generic such that they can be consumed by other projects besides CrosVM and Firecracker. One example that is often mentioned is building a container specific VMM. By doing so, the container VMM can reuse most of the rust-vmm components and simply build the glue around them as well as an appropriate API for interacting with the container.
- **Security & Testability**. Having independent components makes fuzz testing easy to apply to each individual package. Our top priority is now vm-virtio as it provides a virtio device implementation. We want to keep a high standard in terms of testing as these virtualization packages are going to be used in production by multiple projects. Each component is individually tested with a set of common build time tests responsible for running unit tests and linters (including coding style checks), and computing the coverage. In the future we are planning to test the integration of the rust-vmm components by providing a reference VMM implementation.
- **Clean interface**. As crates are shared between multiple VMMs, the interface has to be flexible enough to be used by multiple VMMs. There are a few rounds of design reviews up to the point we are confident the interface is clean and reusable by other projects.

Status:

Each rust-vmm crate lives in its own repository. Depending on where the latest crate code is, the crate can be in one of 3 states:

1. `empty` : The crate repo exists but there is no code there yet. This means the crate was created after a community acknowledgement about its relevance to the project, but either no code has been submitted yet or the initial code base is being reviewed through a pull request.
2. `rust-vmm` : The code is a work in progress. Once it meets the production requirements, the crate will be pushed to crates.io.
3. `crates.io` : The crate is considered to be production ready. It can be consumed from Rust's canonical crate registry: crates.io. The crate repository in rust-vmm is as a staging area until the next push to crates.io. In other words, the production ready version of the code lives in crates.io while development happens in the rust-vmm git repo.

empty

These are the empty repositories that have PRs waiting to be merged.

- vmm-vcpu: a hypervisor-agnostic abstraction for Virtual CPUs (vCPUs).

rust-vmm

- vm-device: a virtual machine device model crate.
- vm-virtio: virtio device trait and implementation for virtio primitives such as virtqueues and descriptor chain.
- vmm-reference: reference VMM built solely with `rust-vmm` crates and minimal glue.
- vhost: a crate to support vhost backend drivers for virtio devices.

crates.io

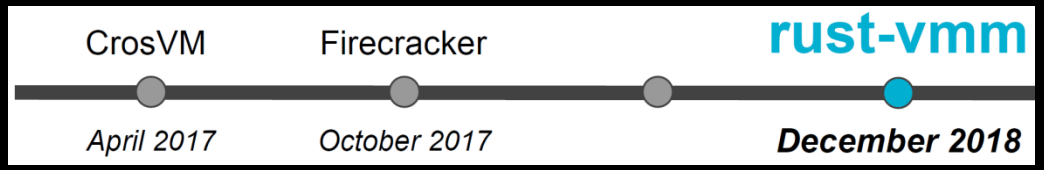
- kvm-bindings: Rust FFI bindings to KVM generated using bindgen.
- kvm-ioctls: Safe wrappers over the KVM API.
- vfio-bindings: Rust FFI bindings for using the VFIO framework.
- virtio-bindings: Rust FFI bindings to virtio kernel headers generated using bindgen.
- vm-memory: abstractions over a virtual machine's memory.
- vmm-sys-util: collection of modules providing helpers and utilities for building VMMs and hypervisors.
- event-manager: abstractions for implementing event based systems.
- linux-loader: parser and loader for vmlinux and bzImage images as well as some other helpers for kernel commandline.
- vm-superio: emulation for legacy devices.

Rust-VMM in Production

- Firecracker
- Cloud Hypervisor
- Alibaba Cloud Sandbox
- Enarx
- libkrun
- Nydus: Dragonfly Container Image Service
- ...

Source: “Rust-vmm Status Report” , Andreea Florescu(Amazon), KVM Forum 2020.

History

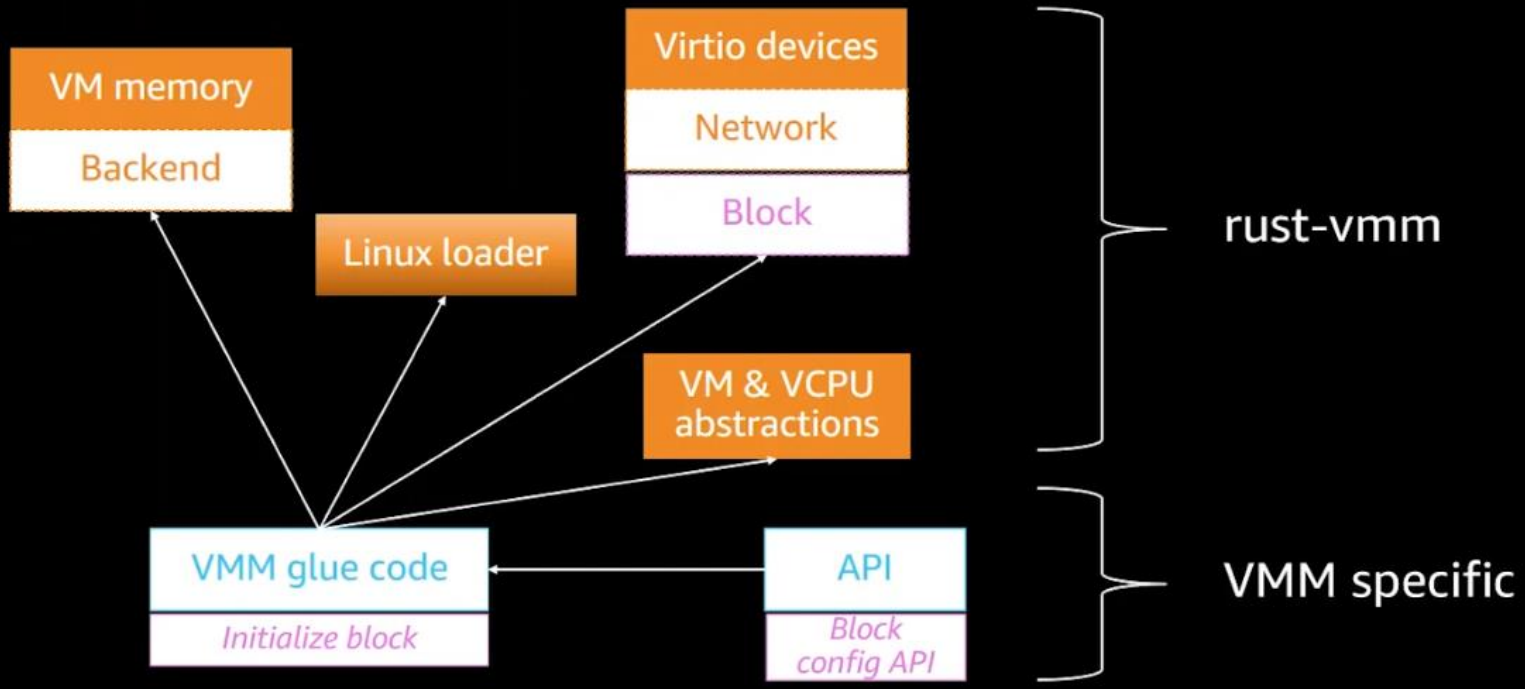


Source: “rust-vmm: shared virtualization crates” , Andreea Florescu(AWS), Fosdem 2019.

Sample

- **HTTP server + storage**

REQUIRED CHANGES



Source: "Rust-vmm: Secure VM-based isolation made simple" , Andreea Florescu(AWS) & Samuel Ortiz(Intel), AWS re:Invent 2020.

3) Cloud-Hypervisor

- **https://github.com/cloud-hypervisor**

Cloud Hypervisor is an open source Virtual Machine Monitor (VMM) that runs on top of KVM and the MSHV hypervisors .

The project focuses on exclusively running modern, cloud workloads, on top of a limited set of hardware architectures and platforms. Cloud workloads refers to those that are usually run by customers inside a cloud provider. For our purposes this means modern operating systems with most I/O handled by paravirtualised devices (i.e. virtio), no requirement for legacy devices, and 64-bit CPUs.

Cloud Hypervisor is implemented in Rust and is based on the rust-vmm crates.

- **Objectives:**

High Level

- Runs on KVM or MSHV
- Minimal emulation
- Low latency
- Low memory footprint
- Low complexity
- High performance
- Small attack surface
- 64-bit support only
- CPU, memory, PCI hotplug
- Machine to machine migration

- **Status:**
  currently supports the x86-64 and AArch64 architectures, and 64-bit Linux and Windows 10/Windows Server 2019 as Guest OS.

Cloud Hypervisor is under active development. The following stability guarantees are currently made:

- The API (including command line options) will not be removed or changed in a breaking way without a minimum of 2 releases notice. Where possible warnings will be given about the use of deprecated functionality and the deprecations will be documented in the release notes.
- Point releases will be made between individual releases where there are substantial bug fixes or security issues that need to be fixed.

Currently the following items are **not** guaranteed across updates:

- Snapshot/restore is not supported across different versions
- Live migration is not supported across different versions
- The following features are considered experimental and may change substantially between releases: TDX, SGX.

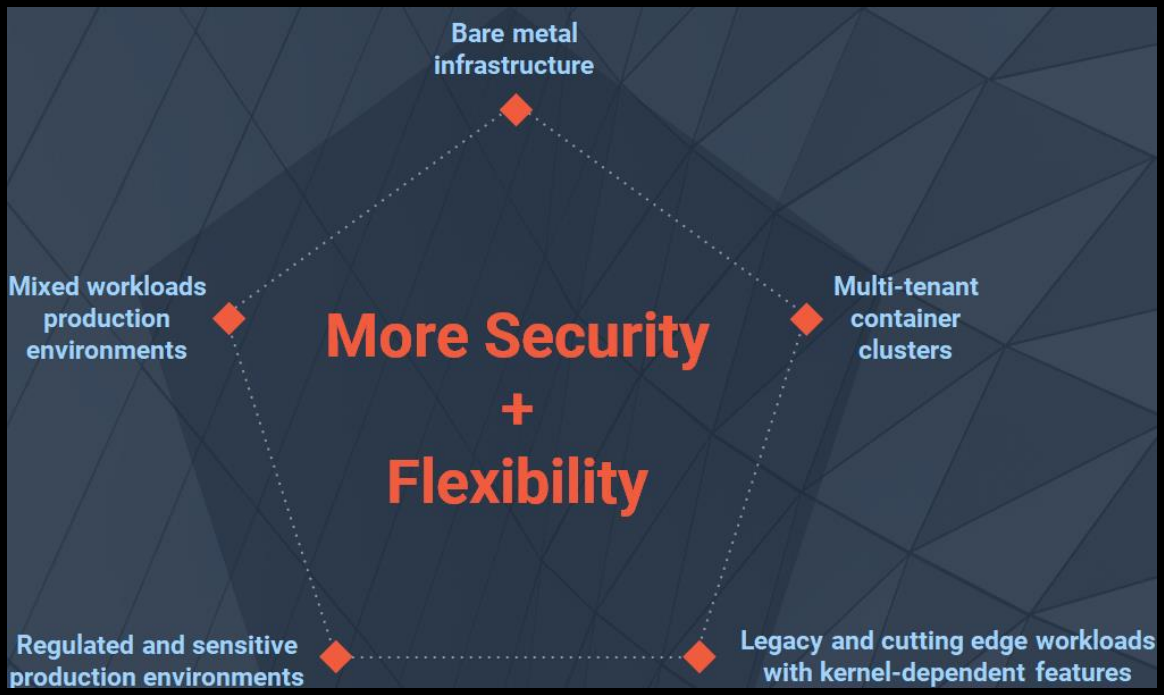
As of 2021-04-29, the following cloud images are supported:

- Ubuntu Bionic (cloudimg)
- Ubuntu Focal (cloudimg)
- Ubuntu Groovy (cloudimg)
- Ubuntu Hirsute (cloudimg)

Direct kernel boot to userspace should work with a rootfs from most distributions.

4) Kata Containers

- https://katacontainers.io/
- A standard implementation of lightweight Virtual Machines (VMs) that feel and perform like containers, but provide the workload isolation and security advantages of VMs.
-

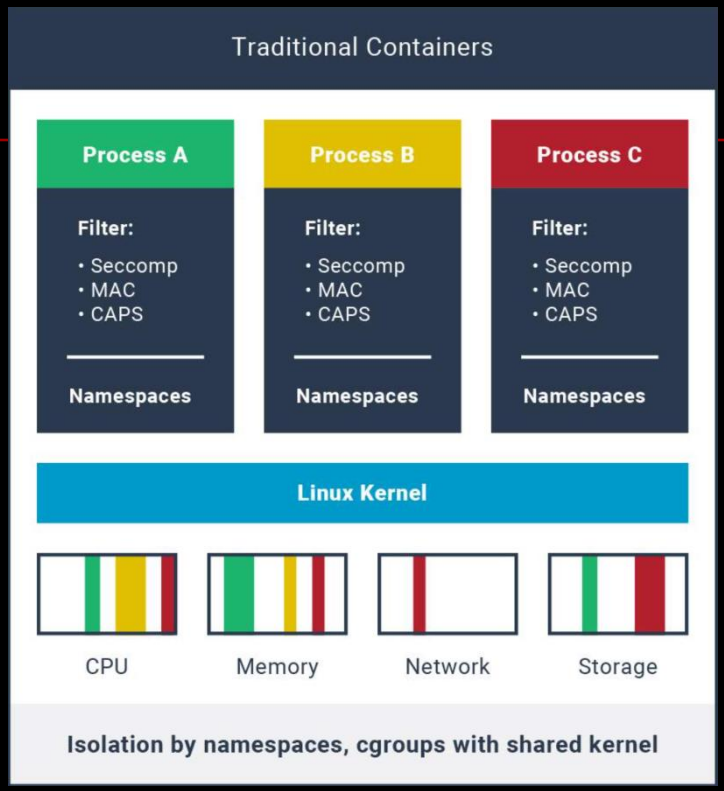


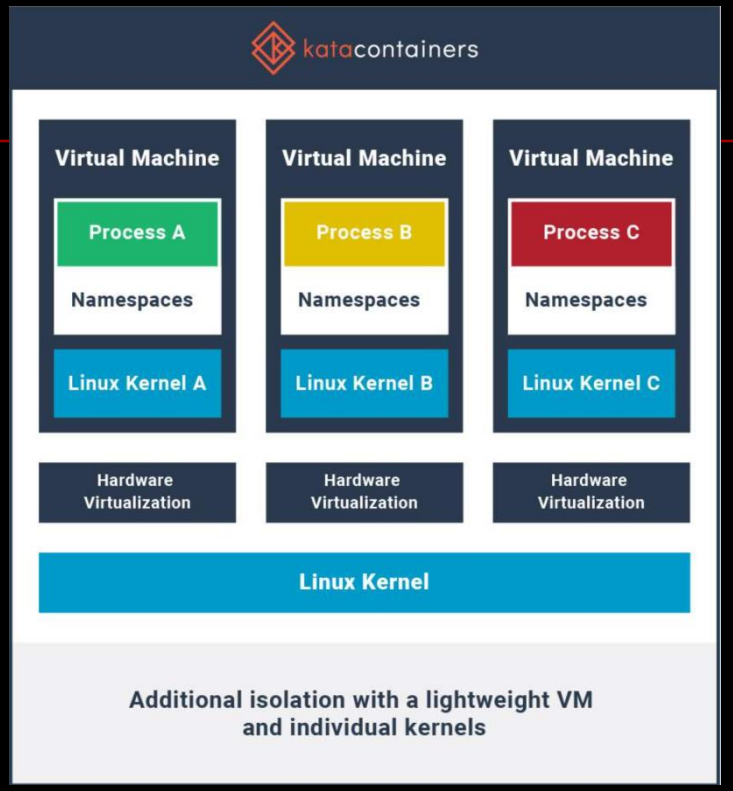
Source: https://katacontainers.io/collateral/kata-containers-onboarding-deck.pptx

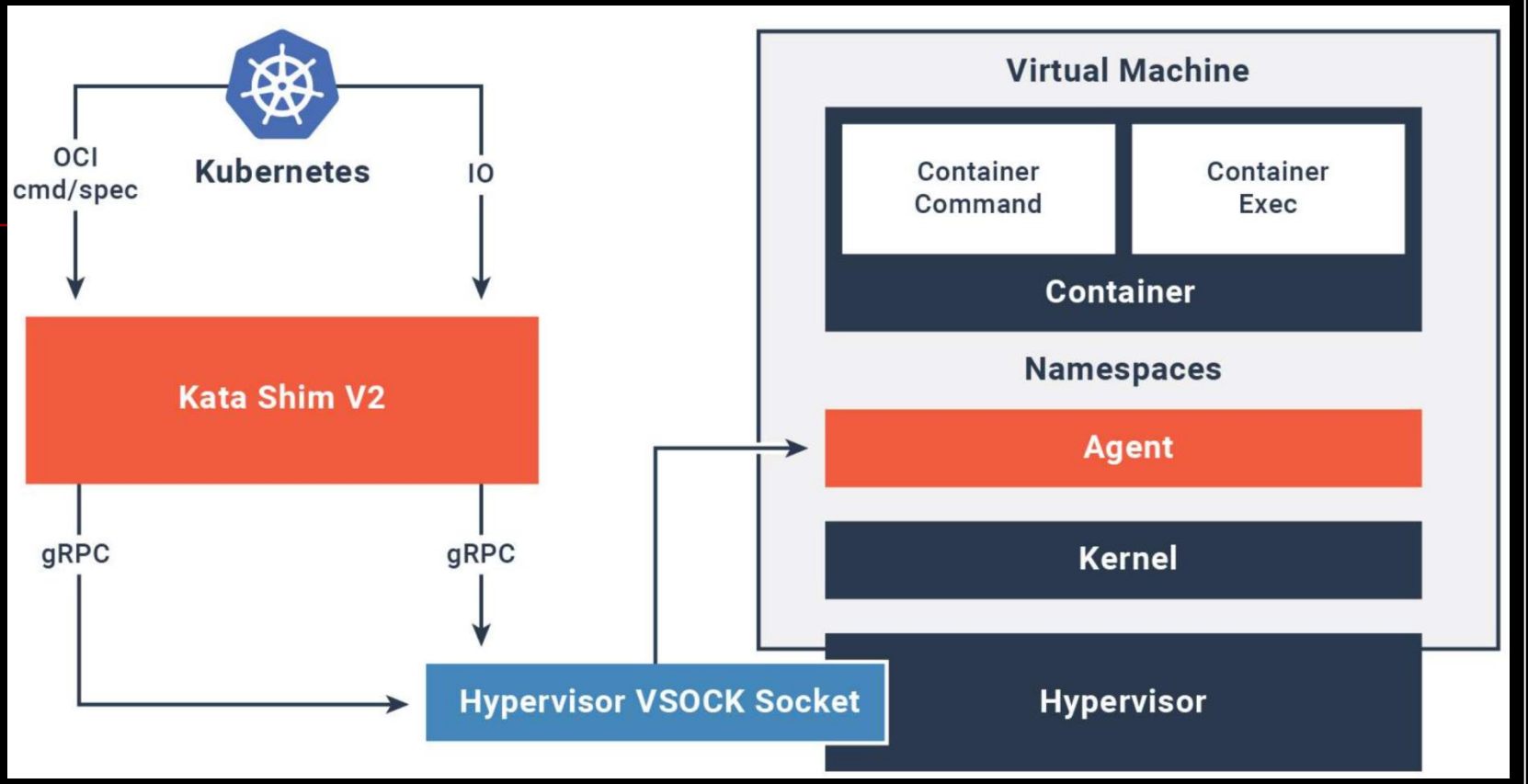
- https://medium.com/kata-containers/kata-containers-2020-annual-report-c9ad7f44e0d4

Architecture & Design

- **https://katacontainers.io/learn/**

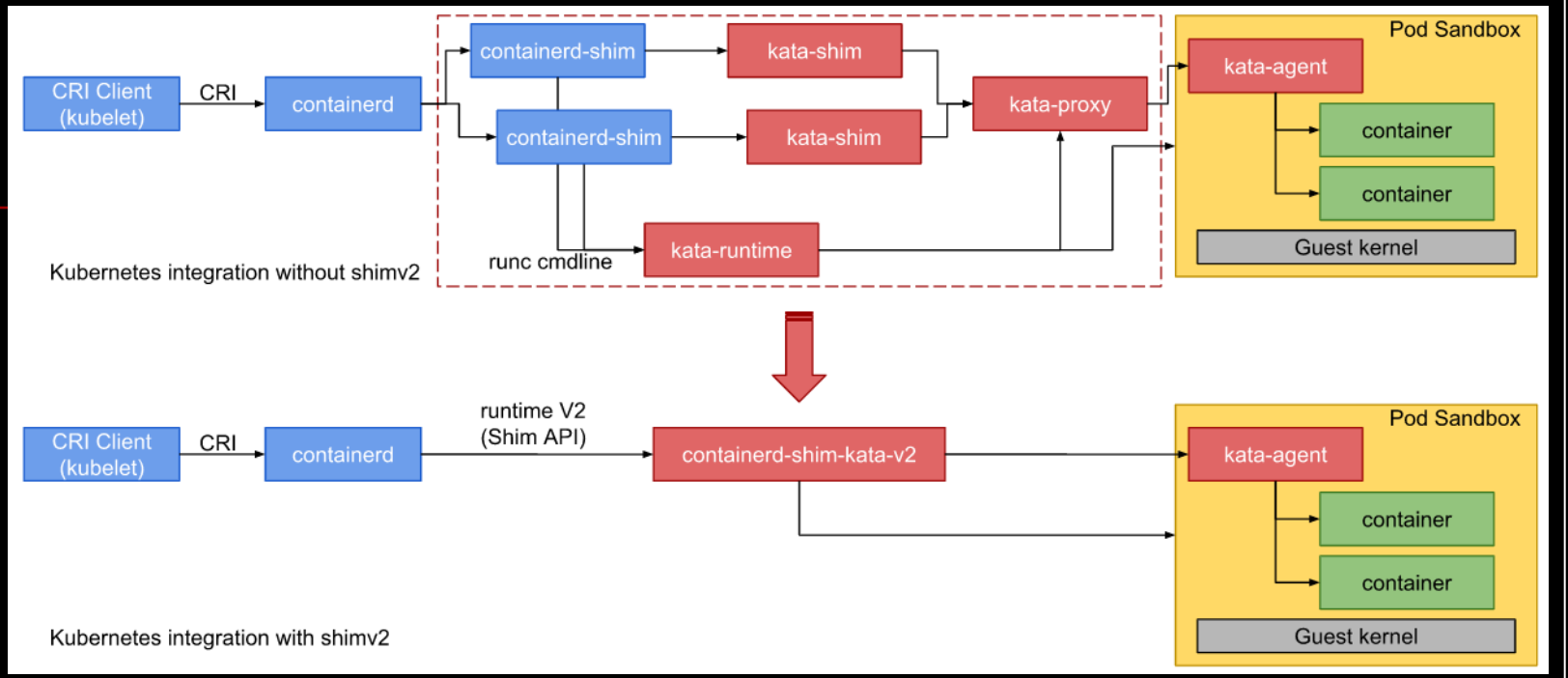






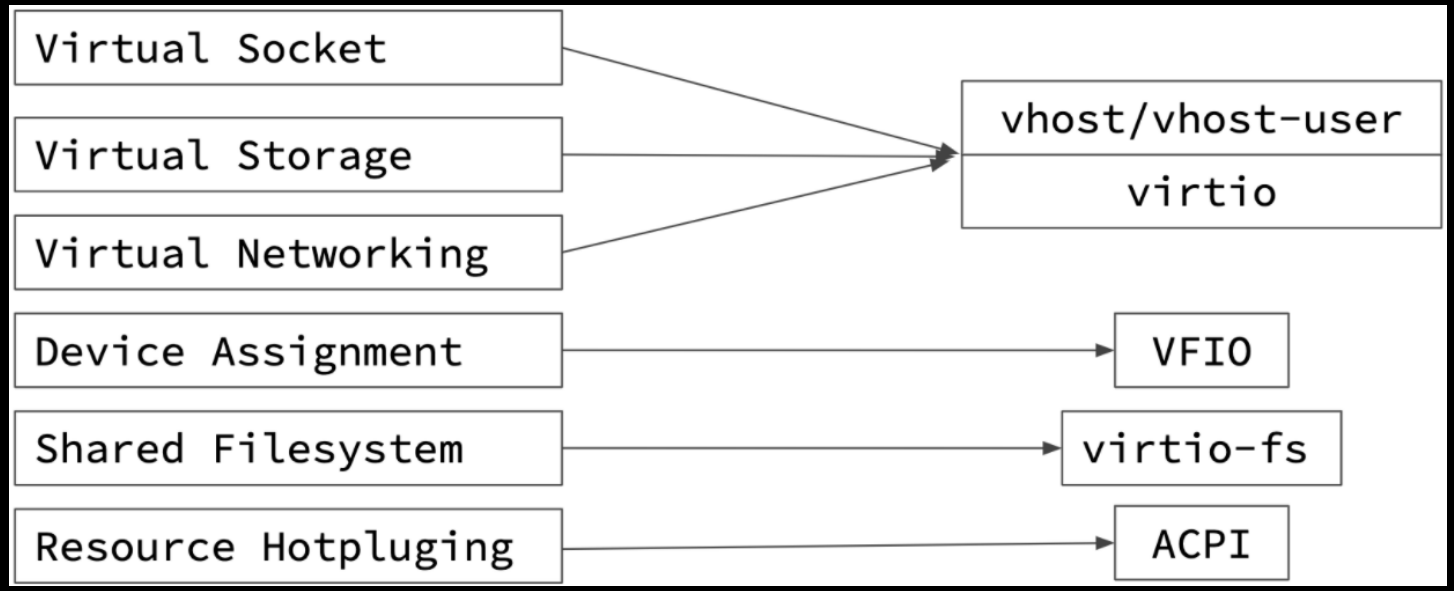
- https://github.com/kata-containers/documentation/blob/master/design/architecture.md

- **from version 1.x to 2.x:**



Virtualization

- **https://github.com/kata-containers/documentation/blob/master/design/virtualization.md**
- **Hypervisor/VMM support:**
  - ACRN hypervisor
  - Cloud Hypervisor/KVM
  - Firecracker/KVM
  - QEMU/KVM
- **Each hypervisor/VMM varies on how or if it handles each of the specific para-virtualized devices or virtualization technologies as below:**

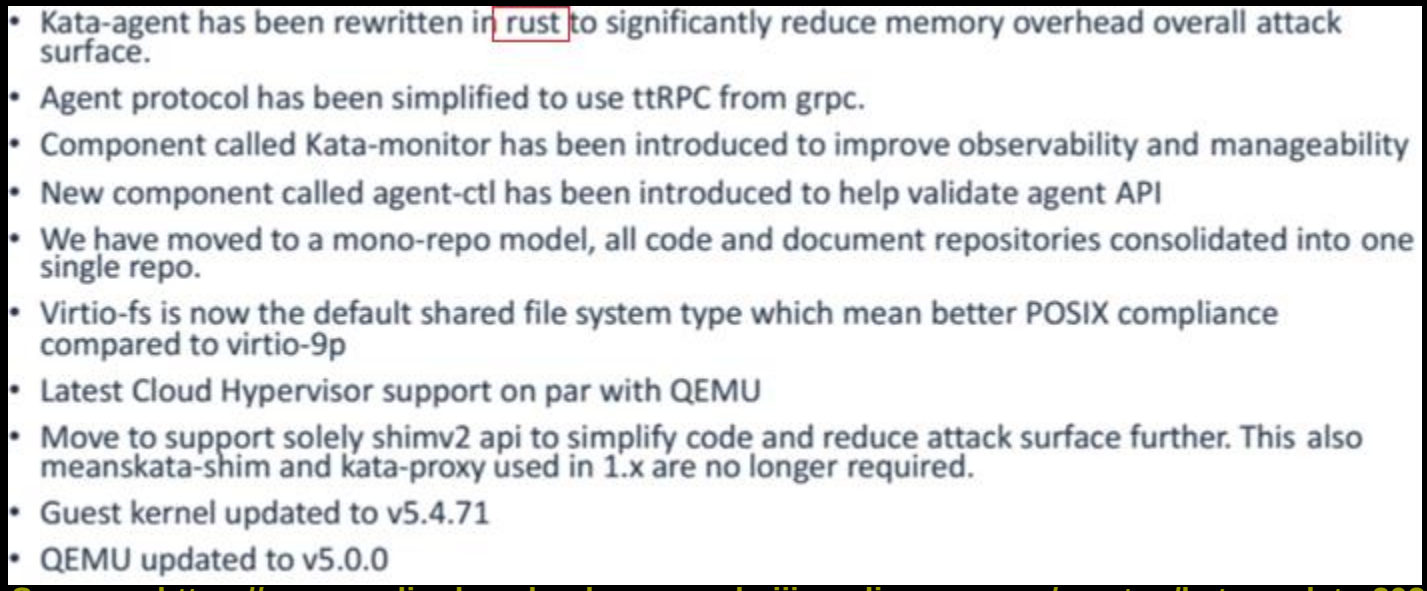


Cloud Native

- Kata Containers runtime is OCI-compatible, CRI-O-compatible, and Containerd-compatible(CNI, CSI…), allowing it to work seamlessly with both Docker and Kubernetes respectively
- https://github.com/kata-containers/documentation/blob/master/how-to/how-to-use-k8s-with-cri-containerd-and-kata.md
-

2.x

- https://github.com/kata-containers/kata-containers/releases/tag/2.0.0
- https://medium.com/kata-containers/kata-containers-version-2-0-e45df4dd328
- What's New



- Kata-agent has been rewritten in rust to significantly reduce memory overhead overall attack surface.
- Agent protocol has been simplified to use ttRPC from grpc.
- Component called Kata-monitor has been introduced to improve observability and manageability
- New component called agent-ctl has been introduced to help validate agent API
- We have moved to a mono-repo model, all code and document repositories consolidated into one single repo.
- Virtio-fs is now the default shared file system type which mean better POSIX compliance compared to virtio-9p
- Latest Cloud Hypervisor support on par with QEMU
- Move to support solely shimv2 api to simplify code and reduce attack surface further. This also meanskata-shim and kata-proxy used in 1.x are no longer required.
- Guest kernel updated to v5.4.71
- QEMU updated to v5.0.0

Source: https://openanolis-downloads.oss-cn-beijing.aliyuncs.com/meetup/kata-update-2020-11.pdf

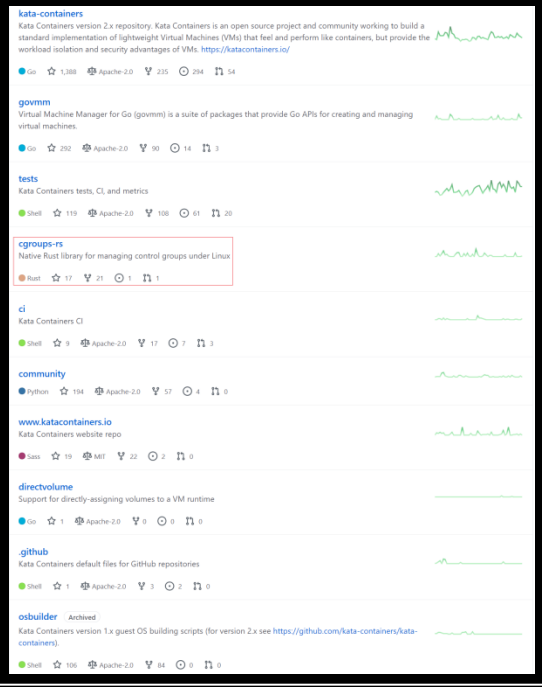
- https://www.openstack.org/videos/summits/virtual/Observability-in-Kata-containers-2.0

- **Ships with Rust Agent**

https://medium.com/kata-containers/kata-containers-version-2-0-e45df4dd328

Kata Containers Version 2.0 Ships with Rust Agent for Improved Security, Performance With 10x Footprint Reduction…

- One of the most fundamental changes is a rewrite of the Kata Containers agent. To help reduce the attack surface and reduce memory overhead, the agent was rewritten in Rust. The main benefit users will see is a 10-fold improvement in size, from 11MB to 300KB. This rewrite and refactoring also introduces utilizing ttRPC, ttRPC moving away from GRPC. ttRPC further reduces the binary size and protocol overhead.



https://github.com/kata-containers/agent

```
[mydev@fedora Bak]$ tree -L 2 kata-containers-main/src/agent
kata-containers-main/src/agent
├── Cargo.lock
├── Cargo.toml
├── kata-agent.service.in
├── kata-containers.target
├── LICENSE
├── Makefile
├── oci
│   ├── Cargo.toml
│   └── src
├── protocols
│   ├── build.rs
│   ├── Cargo.toml
│   ├── hack
│   ├── protos
│   └── src
├── README.md
├── rustjail
│   ├── Cargo.toml
│   └── src
├── src
│   ├── ccw.rs
│   ├── config.rs
│   ├── console.rs
│   ├── device.rs
│   ├── linux_abi.rs
│   ├── main.rs
│   ├── metrics.rs
│   ├── mount.rs
│   ├── namespace.rs
│   ├── netlink.rs
│   ├── network.rs
│   ├── pci.rs
│   ├── random.rs
│   ├── rpc.rs
│   ├── sandbox.rs
│   ├── signal.rs
│   ├── test_utils.rs
│   ├── tracer.rs
│   ├── uevent.rs
│   ├── util.rs
│   ├── version.rs.in
│   └── watcher.rs
├── VERSION -> ../../VERSION
└── vsock-exporter
    ├── Cargo.toml
    └── src
```

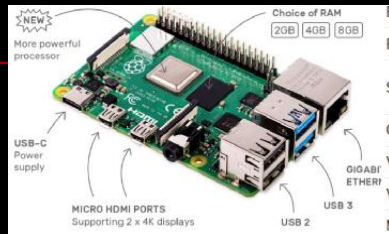
Getting Started

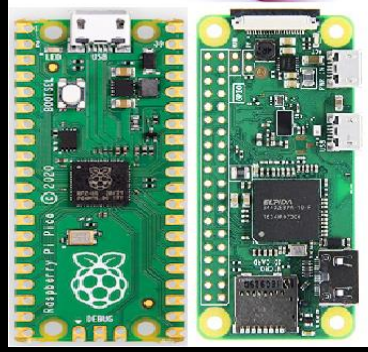
- https://github.com/kata-containers/kata-containers#getting-started
- https://github.com/kata-containers/kata-containers/tree/main/docs
- https://github.com/kata-containers/kata-containers/tree/main/docs/install
- https://github.com/kata-containers/kata-containers/blob/main/docs/install/README.md#packaged-installation-methods
- https://docs.01.org/clearlinux/latest/tutorials/kata.html
- https://github.com/kata-containers/kata-containers/tree/main/docs/install#build-from-source-installation
- https://github.com/kata-containers/documentation/blob/master/how-to/containerd-kata.md
- …

5) Testbed

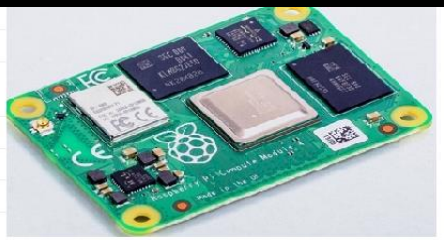
Raspberry Pi

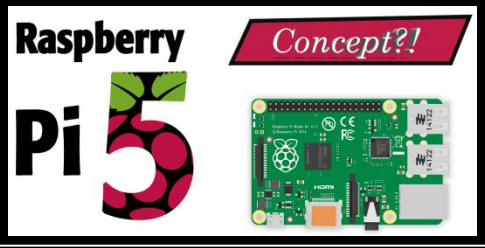
- **https://www.raspberrypi.org/products/raspberry-pi-4-model-b/**





Features/Specs	Raspberry Pi 4B	Raspberry Pi 3 B+
Release date	24th June 2019	14th March 2018
SoC	Broadcom BCM2711 quad-core Cortex-A72 @ 1.5 GHz	Broadcom BCM2837B0 quad-core Cortex-A53 @ 1.4 GHz
GPU	VideoCore VI with OpenGL ES 1.1, 2.0, 3.0	VideoCore IV with OpenGL ES 1.1, 2.0
Video Decode	H.265 4Kp60, H.264 1080p60	H.264 & MPEG-4 1080p30
Video Encode	H.264 1080p30	
Memory	1GB, 2GB, or 4GB LPDDR4	1GB LPDDR2
Storage	microSD card	
Video & Audio Output	2x micro HDMI ports up to 4Kp60 3.5mm AV port (composite + audio) MIPI DSI connector	1x HDMI 1.4 port up to 1080p60 3.5mm AV port (composite + audio) MIPI DSI connector
Camera	MIPI CSI connector	
Ethernet	Native Gigabit Ethernet	Gigabit Ethernet over USB (300 Mbps max.)
WiFi	Dual band 802.11 b/g/n/ac	
Bluetooth	Bluetooth 5.0 + BLE	Bluetooth 4.2 + BLE
USB	2x USB 3.0 + 2x USB 2.0	4x USB 2.0
Expansion	40-pin GPIO header	
Power Supply	5V via USB type-C up to 3A 5V via GPIO header up to 3A Power over Ethernet via PoE HAT	5V via micro USB up to 2.5A 5V via GPIO header up to 3A Power over Ethernet via PoE HAT
Dimensions	85×56 mm	
Default OS	Raspbian (after June 24, 2019)	Raspbian (after March 2018)
Price	$35 (1GB RAM), $45 (2GB RAM), $55 (4GB RAM)	$35 (1GB RAM)



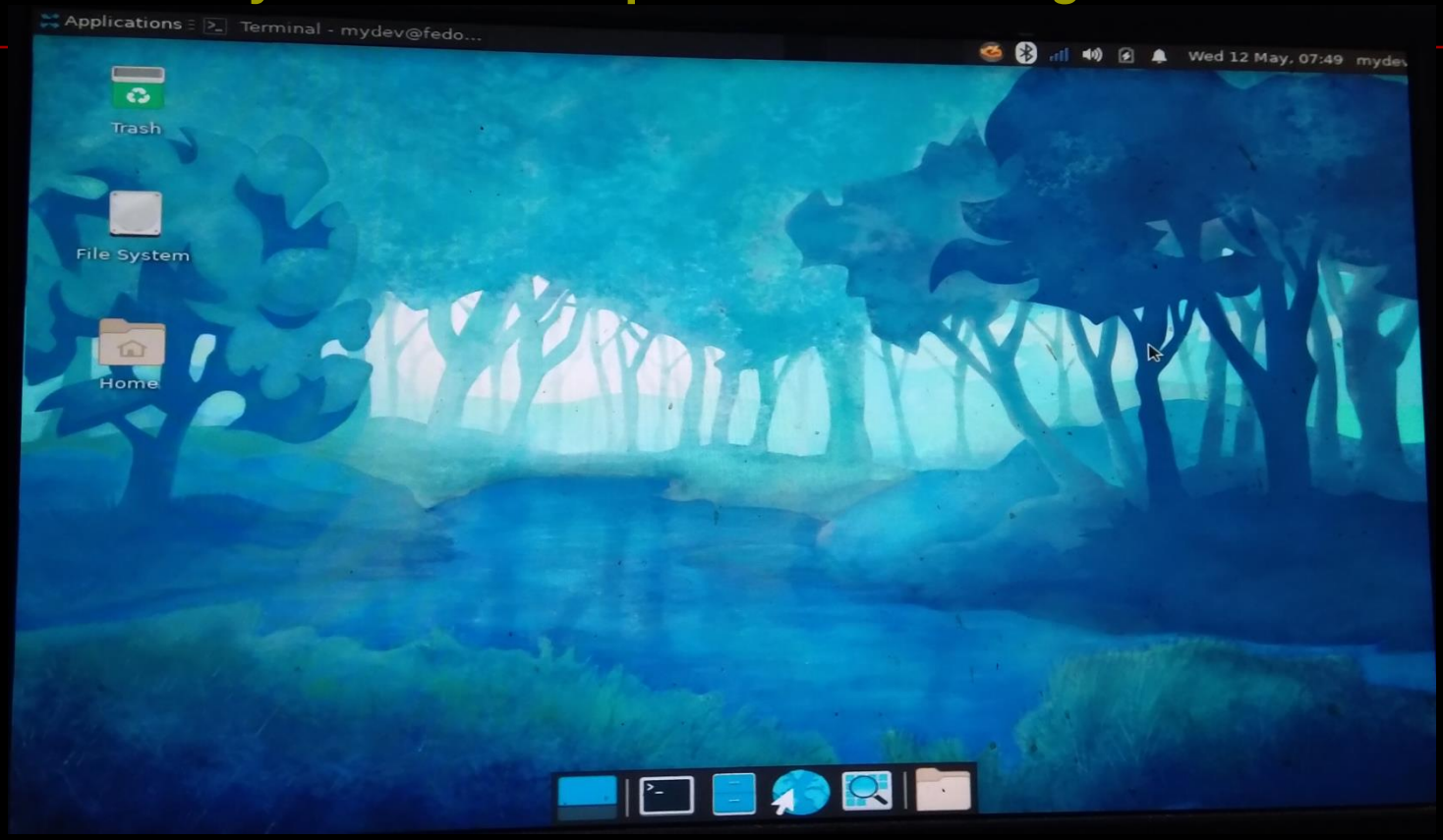


5.1 Fedora

- A Linux distribution developed by the community-supported Fedora Project which is sponsored primarily by Red Hat
- https://en.wikipedia.org/wiki/Fedora_(operating_system)
- https://getfedora.org/
- https://alt.fedoraproject.org/alt/
- https://spins.fedoraproject.org/
- https://fedoraproject.org/wiki/Architectures/ARM
- https://fedoramagazine.org/
- https://silverblue.fedoraproject.org/
- Developer friendly!

Fedora 34 AArch64 XFCE

- **https://fedoramagazine.org/announcing-fedora-34/**
- **https://archives.fedoraproject.org/pub/fedora-secondary/releases/34/Spins/aarch64/images/**



Kernel configuration

```
[mydev@fedora /]$ cat /boot/config-5.13.7-200.fc34.aarch64 |grep -i kvm
CONFIG_KVM=y
CONFIG_HAVE_KVM_IRQCHIP=y
CONFIG_HAVE_KVM_IRQFD=y
CONFIG_HAVE_KVM_IRQ_ROUTING=y
CONFIG_HAVE_KVM_EVENTFD=y
CONFIG_KVM_MMIO=y
CONFIG_HAVE_KVM_MSI=y
CONFIG_HAVE_KVM_CPU_RELAX_INTERCEPT=y
CONFIG_KVM_VFIO=y
CONFIG_HAVE_KVM_ARCH_TLB_FLUSH_ALL=y
CONFIG_KVM_GENERIC_DIRTYLOG_READ_PROTECT=y
CONFIG_HAVE_KVM_IRQ_BYPASS=y
CONFIG_HAVE_KVM_VCPU_RUN_PID_CHANGE=y
CONFIG_PTP_1588_CLOCK_KVM=m
[mydev@fedora /]$
[mydev@fedora /]$ cat /boot/config-5.13.7-200.fc34.aarch64 |grep -i virtual
CONFIG_VIRTUALIZATION=y
# CONFIG_NFC_VIRTUAL_NCI is not set
# Virtual GPIO drivers
# end of Virtual GPIO drivers
CONFIG_REGULATOR_VIRTUAL_CONSUMER=m
CONFIG_FB_VIRTUAL=m
CONFIG_DMA_VIRTUAL_CHANNELS=y
CONFIG_ARCH_HAS_DEBUG_VIRTUAL=y
# CONFIG_DEBUG_VIRTUAL is not set
[mydev@fedora /]$
```

```
[mydev@fedora /]$ cat /boot/config-5.13.7-200.fc34.aarch64 |grep -i cgroup
CONFIG_CGROUPS=y
CONFIG_BLK_CGROUP=y
CONFIG_CGROUP_WRITEBACK=y
CONFIG_CGROUP_SCHED=y
CONFIG_CGROUP_PIDS=y
# CONFIG_CGROUP_RDMA is not set
CONFIG_CGROUP_FREEZER=y
# CONFIG_CGROUP_HUGETLB is not set
CONFIG_CGROUP_DEVICE=y
CONFIG_CGROUP_CPUACCT=y
CONFIG_CGROUP_PERF=y
CONFIG_CGROUP_BPF=y
CONFIG_CGROUP_MISC=y
# CONFIG_CGROUP_DEBUG is not set
CONFIG_SOCK_CGROUP_DATA=y
CONFIG_BLK_CGROUP_RWSTAT=y
CONFIG_BLK_CGROUP_IOLATENCY=y
CONFIG_BLK_CGROUP_IOCOST=y
# CONFIG_BFQ_CGROUP_DEBUG is not set
CONFIG_NETFILTER_XT_MATCH_CGROUP=m
CONFIG_NET_CLS_CGROUP=y
CONFIG_CGROUP_NET_PRIO=y
CONFIG_CGROUP_NET_CLASSID=y
[mydev@fedora /]$
[mydev@fedora /]$ cat /boot/config-5.13.7-200.fc34.aarch64 |grep -i bpf
CONFIG_BPF=y
CONFIG_HAVE_EBPF_JIT=y
CONFIG_ARCH_WANT_DEFAULT_BPF_JIT=y
# BPF subsystem
CONFIG_BPF_SYSCALL=y
CONFIG_BPF_JIT=y
CONFIG_BPF_JIT_ALWAYS_ON=y
CONFIG_BPF_JIT_DEFAULT_ON=y
CONFIG_BPF_UNPRIV_DEFAULT_OFF=y
CONFIG_BPF_PRELOAD=y
CONFIG_BPF_PRELOAD_UMD=m
CONFIG_BPF_LSM=y
# end of BPF subsystem
CONFIG_CGROUP_BPF=y
CONFIG_IPV6_SEG6_BPF=y
CONFIG_NETFILTER_XT_MATCH_BPF=m
# CONFIG_BPFILTER is not set
CONFIG_NET_CLS_BPF=m
CONFIG_NET_ACT_BPF=m
CONFIG_BPF_STREAM_PARSER=y
CONFIG_LWTUNNEL_BPF=y
CONFIG_BPF_LIRC_MODE2=y
CONFIG_LSM="lockdown,yama,integrity,selinux,bpf,landlock"
CONFIG_BPF_EVENTS=y
# CONFIG_BPF_KPROBE_OVERRIDE is not set
# CONFIG_TEST_BPF is not set
[mydev@fedora /]$
```

5.2 Manjaro

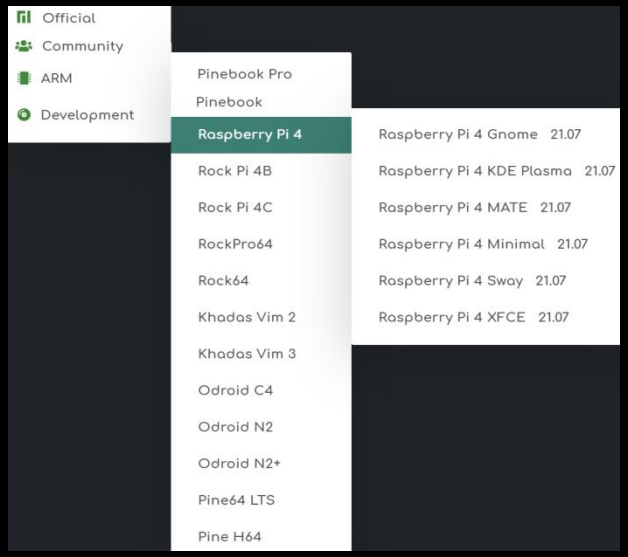
- An open-source Linux distribution based on the Arch Linux operating system
- https://en.wikipedia.org/wiki/Manjaro_Linux
- https://distrowatch.com/

Page Hit Ranking

Data span: Last 6 months Go

Rank	Distribution	HPD*
1	MX Linux	3107▼
2	Manjaro	2309▼
3	EndeavourOS	2080▲
4	Mint	1996▼
5	Pop!_OS	1697▼
6	Ubuntu	1332▼
7	Debian	1202▲
8	elementary	1070▼
9	Garuda	1006▲
10	Fedora	984▼

- https://manjaro.org



- Much stable than Fedora on RPi4, and More and more developer friendly!

- **But:**

```
[myrpi4@manjarorpi4node1 ~]$ uname -a
Linux manjarorpi4node1 5.12.17-2-MANJARO-ARM #1 SMP PREEMPT Fri Jul 16 18:21:07 CDT 2021 aarch64 GNU/Linux
```

and no available official package for Kata Containers.

For details, you may refer to my previous talk "Kata Containers for Edge Computing" at Kata Containers Meetup(Shanghai 2021).

5.3 Dev Env

RPi4 with Fedora 34

- sudo dnf install rust cargo golang kata-runtime libfdt-devel…
-

```
[mydev@fedora ~]$ uname -a
Linux fedora 5.13.7-200.fc34.aarch64 #1 SMP Sat Jul 31 13:54:43 UTC 2021 aarch64 aarch64 aarch64 GNU/Linux
[mydev@fedora ~]$
[mydev@fedora ~]$ gcc -v
Using built-in specs.
COLLECT_GCC=/usr/bin/gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/aarch64-redhat-linux/11/lto-wrapper
Target: aarch64-redhat-linux
Configured with: ../configure --enable-bootstrap --enable-languages=c,c++,fortran,objc,obj-c++,ada,go,lto --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=http://bugzill
a.redhat.com/bugzilla --enable-shared --enable-threads=posix --enable-checking=release --enable-multilib --with-system-zlib --enable-__cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-o
bject --enable-linker-build-id --with-gcc-major-version-only --with-linker-hash-style=gnu --enable-plugin --enable-initfini-array --with-isl=/builddir/build/BUILD/gcc-11.2.1-20210728/obj-aarch64-red
hat-linux/isl-install --enable-gnu-indirect-function --build=aarch64-redhat-linux
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 11.2.1 20210728 (Red Hat 11.2.1-1) (GCC)
[mydev@fedora ~]$
[mydev@fedora ~]$ clang -v
clang version 12.0.0 (Fedora 12.0.0-2.fc34)
Target: aarch64-unknown-linux-gnu
Thread model: posix
InstalledDir: /usr/bin
Found candidate GCC installation: /usr/bin/../lib/gcc/aarch64-redhat-linux/11
Found candidate GCC installation: /usr/lib/gcc/aarch64-redhat-linux/11
Selected GCC installation: /usr/lib/gcc/aarch64-redhat-linux/11
Candidate multilib: .;@m64
Selected multilib: .;@m64
[mydev@fedora ~]$
[mydev@fedora ~]$ rustc -V
rustc 1.53.0 (Fedora 1.53.0-1.fc34)
[mydev@fedora ~]$
[mydev@fedora ~]$ cargo -V
cargo 1.53.0
[mydev@fedora ~]$
[mydev@fedora ~]$ go version
go version go1.16.7 linux/arm64
[mydev@fedora ~]$
[mydev@fedora ~]$ kata-runtime -v
kata-runtime  : 2.1.0
   commit   : 04c0183bceafe68bae317809de6aad0d47679dd1
   OCI specs: 1.0.1-dev
[mydev@fedora ~]$
```

```
[mydev@fedora /]$ docker -v
Docker version 20.10.8, build 3967b7d
[mydev@fedora /]$
[mydev@fedora /]$ podman -v
podman version 3.2.3
[mydev@fedora /]$
[mydev@fedora /]$ containerd -v
containerd containerd.io 1.4.9 e25210fe30a0a703442421b0f60afac609f950a3
[mydev@fedora /]$
```

```
[mydev@fedora ~]$ dmesg |grep -i kvm
[    1.064777] kvm [1]: IPA Size Limit: 44 bits
[    1.069791] kvm [1]: vgic interrupt IRQ9
[    1.072657] kvm [1]: Hyp mode initialized successfully
[mydev@fedora ~]$
```

II. Cloud-Hypervisor + Kata Containers

1) Cloud-Hypervisor on RPi4

1.1 Build & Test

- https://github.com/cloud-hypervisor/cloud-hypervisor
- https://github.com/cloud-hypervisor/cloud-hypervisor/blob/master/docs/arm64.md
- https://www.cnblogs.com/dream397/p/13844690.html
- https://medium.com/@michael2012zhao_67085/cloud-hypervisor-on-raspberry-pi-4b-8042a24c5013

Setup Cargo

-
```
[mydev@fedora /]$ cat ~/.cargo/config
[source.crates-io]
registry = "https://github.com/rust-lang/crates.io-index"
replace-with = 'ustc'

[source.ustc]
registry = "https://mirrors.ustc.edu.cn/crates.io-index"

[source.sjtu]
registry = "https://mirrors.sjtug.sjtu.edu.cn/git/crates.io-index/"

[source.tuna]
registry = "https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git"

[source.rustcc]
registry = "https://code.aliyun.com/rustcc/crates.io-index.git"
[mydev@fedora /]$
```

Non-Containerised

- **cargo build --no-default-features --features mmio,kvm**

```
    Updating `https://mirrors.ustc.edu.cn/crates.io-index` index
    Updating git repository `https://github.com/rust-vmm/vfio-ioctls`
    Updating git submodule `https://github.com/rust-vmm/rust-vmm-ci.git`
    Updating git repository `https://github.com/firecracker-microvm/firecracker`
    Updating git repository `https://github.com/firecracker-microvm/micro-http`
    Updating git submodule `https://github.com/rust-vmm/rust-vmm-ci.git`
error: Package `cloud-hypervisor v17.0.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master)` does not have the feature `mmio`
```

https://en.wikipedia.org/wiki/Memory-mapped_I/O

- **cargo build --no-default-features –features kvm**

```
Downloading crates ...
 Downloaded ansi_term v0.11.0 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded kvm-ioctls v0.9.0 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded lazy_static v1.4.0 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded libc v0.2.98 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded log v0.4.14 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded proc-macro2 v1.0.28 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded linux-loader v0.3.0 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded quote v1.0.9 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded remain v0.2.2 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded ryu v1.0.5 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded serde v1.0.127 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded serde_derive v1.0.127 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded serde_json v1.0.66 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded signal-hook v0.3.9 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded signal-hook-registry v1.4.0 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded strsim v0.8.0 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded syn v1.0.74 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded term_size v0.3.2 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded textwrap v0.11.0 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded thiserror v1.0.26 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded thiserror-impl v1.0.26 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded unicode-width v0.1.8 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded unicode-xid v0.2.2 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded uuid v0.8.2 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded vec_map v0.8.2 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded versionize v0.1.6 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded vfio-bindings v0.2.0 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded vhost v0.1.0 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded virtio-bindings v0.1.0 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded vm-memory v0.5.0 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded vm-memory v0.6.0 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded vmm-sys-util v0.8.0 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded anyhow v1.0.42 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded arc-swap v1.3.0 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded atty v0.2.14 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded bincode v1.3.3 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded bitflags v1.2.1 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded byteorder v1.4.3 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded cfg-if v1.0.0 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded clap v2.33.3 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded crc64 v1.0.0 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded epoll v4.3.1 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded fdt v0.1.3 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded io-uring v0.5.1 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
 Downloaded itoa v0.4.7 (registry `https://mirrors.ustc.edu.cn/crates.io-index`)
  Compiling libc v0.2.98
  Compiling proc-macro2 v1.0.28
```

...

...

```
Compiling signal-hook-registry v1.4.0
Compiling seccomp v0.1.0 (https://github.com/firecracker-microvm/firecracker?tag=v0.24.5#cd36c699)
Compiling textwrap v0.11.0
Compiling linux-loader v0.3.0
Compiling vm-virtio v0.1.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master/vm-virtio)
Compiling clap v2.33.3
Compiling bincode v1.3.3
Compiling cloud-hypervisor v17.0.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master)
Compiling thiserror-impl v1.0.26
Compiling versionize_derive v0.1.4 (https://github.com/cloud-hypervisor/versionize_derive?branch=ch#ae35ef7a)
Compiling remain v0.2.2
Compiling thiserror v1.0.26
Compiling vmm-sys-util v0.8.0
Compiling event_monitor v0.1.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master/event_monitor)
Compiling kvm-bindings v0.4.0 (https://github.com/cloud-hypervisor/kvm-bindings?branch=ch-v0.4.0#1a687256)
Compiling versionize v0.1.6
Compiling vfio-bindings v0.2.0
Compiling rate_limiter v0.1.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master/rate_limiter)
Compiling qcow v0.1.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master/qcow)
Compiling net_gen v0.1.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master/net_gen)
Compiling vhost v0.1.0
Compiling micro_http v0.1.0 (https://github.com/firecracker-microvm/micro-http?branch=main#9517a300)
Compiling api_client v0.1.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master/api_client)
Compiling vm-migration v0.1.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master/vm-migration)
Compiling net_util v0.1.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master/net_util)
Compiling kvm-ioctls v0.9.0
Compiling block_util v0.1.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master/block_util)
Compiling hypervisor v0.1.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master/hypervisor)
Compiling vfio-ioctls v0.1.0 (https://github.com/rust-vmm/vfio-ioctls?branch=master#9b84069e)
Compiling vm-device v0.1.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master/vm-device)
Compiling arch v0.1.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master/arch)
Compiling vm-allocator v0.1.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master/vm-allocator)
Compiling devices v0.1.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master/devices)
Compiling pci v0.1.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master/pci)
Compiling virtio-devices v0.1.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master/virtio-devices)
Compiling vmm v0.1.0 (/opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master/vmm)
 Finished dev [unoptimized + debuginfo] target(s) in 3m 56s
```

successfully built it!

Run

```
[mydev@fedora cloud-hypervisor-master]$ ls -al target/debug/cloud-hypervisor
-rwxr-xr-x. 2 mydev mydev 62331224 Aug  7 03:17 target/debug/cloud-hypervisor
[mydev@fedora cloud-hypervisor-master]$ file target/debug/cloud-hypervisor
target/debug/cloud-hypervisor: ELF 64-bit LSB pie executable, ARM aarch64, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-aarch64.so.1, BuildID[sha1]=bfe1f18df83ad75d6eede0dd9fe86b5
4ca5fa4c5, for GNU/Linux 3.7.0, with debug_info, not stripped
[mydev@fedora cloud-hypervisor-master]$
[mydev@fedora cloud-hypervisor-master]$ ll /opt/MyWorkSpace/Tmp/Virtual/Cloud-Hypervisor
total 0
drwxr-xr-x. 1 mydev mydev  0 Aug  7 04:19 ./
drwxr-xr-x. 1 mydev mydev 80 Aug  7 04:19 ../
[mydev@fedora cloud-hypervisor-master]$
```

```
[mydev@fedora cloud-hypervisor-master]$ cp target/debug/cloud-hypervisor /opt/MyWorkSpace/Tmp/Virtual/Cloud-Hypervisor
[mydev@fedora cloud-hypervisor-master]$ pushd /opt/MyWorkSpace/Tmp/Virtual/Cloud-Hypervisor
/opt/MyWorkSpace/Tmp/Virtual/Cloud-Hypervisor /opt/MyWorkSpace/MyProjs/Virtual/Hypervisor/Cloud-Hypervisor/cloud-hypervisor-master
[mydev@fedora Cloud-Hypervisor]$
[mydev@fedora Cloud-Hypervisor]$ wget https://s3.amazonaws.com/spec.ccfc.min/img/aarch64/ubuntu_with_ssh/fsfiles/xenial.rootfs.ext4
--2021-08-07 04:21:52--  https://s3.amazonaws.com/spec.ccfc.min/img/aarch64/ubuntu_with_ssh/fsfiles/xenial.rootfs.ext4
Resolving s3.amazonaws.com (s3.amazonaws.com)... 52.217.131.176
Connecting to s3.amazonaws.com (s3.amazonaws.com)|52.217.131.176|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 209715200 (200M) [application/x-www-form-urlencoded]
Saving to: 'xenial.rootfs.ext4'

xenial.rootfs.ext4                          100%[=============================================================================================================>] 200.00M  8.70MB/s    in 30s

2021-08-07 04:22:23 (6.70 MB/s) - 'xenial.rootfs.ext4' saved [209715200/209715200]

[mydev@fedora Cloud-Hypervisor]$ wget https://s3.amazonaws.com/spec.ccfc.min/img/aarch64/ubuntu_with_ssh/kernel/vmlinux.bin -O kernel.bin
--2021-08-07 04:33:16--  https://s3.amazonaws.com/spec.ccfc.min/img/aarch64/ubuntu_with_ssh/kernel/vmlinux.bin
Resolving s3.amazonaws.com (s3.amazonaws.com)... 52.216.137.54
Connecting to s3.amazonaws.com (s3.amazonaws.com)|52.216.137.54|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8356352 (8.0M) [application/macbinary]
Saving to: 'kernel.bin'

kernel.bin                                  100%[=============================================================================================================>]   7.97M  2.75MB/s    in 2.9s

2021-08-07 04:33:20 (2.75 MB/s) - 'kernel.bin' saved [8356352/8356352]

[mydev@fedora Cloud-Hypervisor]$ ll
total 273812
drwxr-xr-x. 1 mydev mydev        88 Aug  7 04:33 ./
drwxr-xr-x. 1 mydev mydev        80 Aug  7 04:19 ../
-rwxr-xr-x. 1 mydev mydev  62331224 Aug  7 04:20 cloud-hypervisor*
-rw-r--r--. 1 mydev mydev   8356352 Nov 24  2020 kernel.bin
-rw-r--r--. 1 mydev mydev 209715200 Sep 10  2019 xenial.rootfs.ext4
[mydev@fedora Cloud-Hypervisor]$
[mydev@fedora Cloud-Hypervisor]$ sudo ./cloud-hypervisor --kernel ./kernel.bin --disk path=./xenial.rootfs.ext4 --cmdline "keep_bootcon console=hvc0 reboot=k panic=1 pci=off root=/dev/vda rw" --cpus
 boot=4 --memory size=512M --serial file=serial.log --log-file log.log -vvv
[sudo] password for mydev:
Error booting VM: VmBoot(ConfigureSystem(AArch64Setup(SetupGic(CreateGic(CreateDevice(No such device (os error 19)))))))
[mydev@fedora Cloud-Hypervisor]$
```

- **https://github.com/cloud-hypervisor/cloud-hypervisor/blob/master/arch/src/aarch64/gic/mod.rs**

```rust
/// Initialize a GIC device
fn init_device(vm: &Arc<dyn hypervisor::Vm>) -> Result<Arc<dyn hypervisor::Device>> {
    let mut gic_device = kvm_bindings::kvm_create_device {
        type_: Self::version(),
        fd: 0,
        flags: 0,
    };

    vm.create_device(&mut gic_device)
        .map_err(super::Error::CreateGic)
}
```

retried it according to the official guide

```
[mydev@fedora Cloud-Hypervisor]$ wget https://cloud-images.ubuntu.com/focal/current/focal-server-cloudimg-arm64.img
--2021-08-07 04:48:14--  https://cloud-images.ubuntu.com/focal/current/focal-server-cloudimg-arm64.img
Resolving cloud-images.ubuntu.com (cloud-images.ubuntu.com)... 91.189.91.123, 91.189.91.124, 2001:67c:1562::28, ...
Connecting to cloud-images.ubuntu.com (cloud-images.ubuntu.com)|91.189.91.123|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 527433728 (503M) [application/octet-stream]
Saving to: 'focal-server-cloudimg-arm64.img'

focal-server-cloudimg-arm64.img                 100%[=============================================================================================================>] 503.00M  9.51MB/s    in 56s

2021-08-07 04:49:11 (9.04 MB/s) - 'focal-server-cloudimg-arm64.img' saved [527433728/527433728]

[mydev@fedora Cloud-Hypervisor]$
[mydev@fedora Cloud-Hypervisor]$ qemu-img convert -p -f qcow2 -O raw focal-server-cloudimg-arm64.img focal-server-cloudimg-arm64.raw
    (100.00/100%)
[mydev@fedora Cloud-Hypervisor]$ wget https://github.com/cloud-hypervisor/rust-hypervisor-firmware/releases/download/0.3.2/hypervisor-fw
--2021-08-07 04:50:29--  https://github.com/cloud-hypervisor/rust-hypervisor-firmware/releases/download/0.3.2/hypervisor-fw
Resolving github.com (github.com)... 20.205.243.166
Connecting to github.com (github.com)|20.205.243.166|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-releases.githubusercontent.com/181578726/efcc869e-3328-41dc-82f9-d677c633019e?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20210807%2Fus-east-1%2
Fs3%2Faws4_request&X-Amz-Date=20210807T115031Z&X-Amz-Expires=300&X-Amz-Signature=443723d5178b0c11b545e25aaf6b2c8a726cb6bd4020bf1e34e794e8fc4ee436&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id
=181578726&response-content-disposition=attachment%3B%20filename%3Dhypervisor-fw&response-content-type=application%2Foctet-stream [following]
--2021-08-07 04:50:31--  https://github-releases.githubusercontent.com/181578726/efcc869e-3328-41dc-82f9-d677c633019e?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F2021080
7%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20210807T115031Z&X-Amz-Expires=300&X-Amz-Signature=443723d5178b0c11b545e25aaf6b2c8a726cb6bd4020bf1e34e794e8fc4ee436&X-Amz-SignedHeaders=host&actor_id=0&k
ey_id=0&repo_id=181578726&response-content-disposition=attachment%3B%20filename%3Dhypervisor-fw&response-content-type=application%2Foctet-stream
Resolving github-releases.githubusercontent.com (github-releases.githubusercontent.com)... 185.199.109.154, 185.199.110.154, 185.199.111.154, ...
Connecting to github-releases.githubusercontent.com (github-releases.githubusercontent.com)|185.199.109.154|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85928 (84K) [application/octet-stream]
Saving to: 'hypervisor-fw'

hypervisor-fw                                   100%[=============================================================================================================>]  83.91K   534KB/s    in 0.2s

2021-08-07 04:50:31 (534 KB/s) - 'hypervisor-fw' saved [85928/85928]
```

```
[mydev@fedora Cloud-Hypervisor]$ ll
total 2102156
drwxr-xr-x. 1 mydev mydev        272 Aug  7 04:50 ./
drwxr-xr-x. 1 mydev mydev         80 Aug  7 04:19 ../
-rwxr-xr-x. 1 mydev mydev   62331224 Aug  7 04:20 cloud-hypervisor*
-rw-r--r--. 1 mydev mydev  527433728 Aug  3 15:24 focal-server-cloudimg-arm64.img
-rw-r--r--. 1 mydev mydev 2361393152 Aug  7 04:50 focal-server-cloudimg-arm64.raw
-rw-r--r--. 1 mydev mydev      85928 Jul 23 01:42 hypervisor-fw
-rw-r--r--. 1 mydev mydev    8356352 Nov 24  2020 kernel.bin
-rw-------. 1 root  root        4314 Aug  7 04:35 log.log
-rw-------. 1 root  root           0 Aug  7 04:35 serial.log
-rw-r--r--. 1 mydev mydev  209715200 Sep 10  2019 xenial.rootfs.ext4
[mydev@fedora Cloud-Hypervisor]$
[mydev@fedora Cloud-Hypervisor]$ file ./hypervisor-fw
./hypervisor-fw: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, stripped
[mydev@fedora Cloud-Hypervisor]$
[mydev@fedora Cloud-Hypervisor]$
[mydev@fedora Cloud-Hypervisor]$ sudo setcap cap_net_admin+ep ./cloud-hypervisor
[sudo] password for mydev:
[mydev@fedora Cloud-Hypervisor]$
[mydev@fedora Cloud-Hypervisor]$ sudo ./cloud-hypervisor --kernel ./kernel.bin --disk path=./xenial.rootfs.ext4 --cmdline "keep_bootcon console=hvc0 reboot=k panic=1 pci=off root=/dev/vda rw" --cpus
 boot=4 --memory size=512M --serial file=serial-capset.log --log-file log-capset.log -vvv
Error booting VM: VmBoot(ConfigureSystem(AArch64Setup(SetupGic(CreateGic(CreateDevice(No such device (os error 19)))))))
[mydev@fedora Cloud-Hypervisor]$
[mydev@fedora Cloud-Hypervisor]$ ./cloud-hypervisor --kernel ./kernel.bin --disk path=./xenial.rootfs.ext4 --cmdline "keep_bootcon console=hvc0 reboot=k panic=1 pci=off root=/dev/vda rw" --cpus boot
=4 --memory size=512M --serial file=serial-capset.log --log-file log-capset.log -vvv
Error creating log file: Permission denied (os error 13)
[mydev@fedora Cloud-Hypervisor]$
[mydev@fedora Cloud-Hypervisor]$ ./cloud-hypervisor --kernel ./kernel.bin --disk path=focal-server-cloudimg-arm64.raw --cpus boot=4 --memory size=1024M --net "tap=,mac=,ip=,mask=" --rng
Error booting VM: VmBoot(ConfigureSystem(AArch64Setup(SetupGic(CreateGic(CreateDevice(No such device (os error 19)))))))
```

failed again…

- **root cause:**

https://github.com/cloud-hypervisor/cloud-hypervisor/blob/master/docs/arm64.md

This document introduces how to build and test Cloud Hypervisor on AArch64 servers. Currently Cloud Hypervisor cannot be tested on Raspberry PI. Because on AArch64, Cloud Hypervisor requires GICv3-ITS device for PCIe MSI interrupt handling. But GICv3-ITS has not been equipped on any Raspberry PI product so far.

Now Cloud Hypervisor supports 2 ways of booting on AArch64: UEFI booting and direct-kernel booting. The document covers both of the ways.

All the steps are based on Ubuntu. We use the Ubuntu cloud image for guest VM disk.

```
[mydev@fedora cloud-hypervisor-master]$ tree -L 2 arch/src/aarch64/gic
arch/src/aarch64/gic
├── dist_regs.rs
├── gicv3_its.rs
├── gicv3.rs
├── icc_regs.rs
├── mod.rs
└── redist_regs.rs
```

https://medium.com/@michael2012zhao_67085/cloud-hypervisor-on-raspberry-pi-4b-8042a24c5013

For Virtio devices, you can choose MMIO or PCI as transport option. But on Raspberry Pi 4B, MMIO is the only option. Because PCI devices requires GICv3-ITS for MSI messaging. GICv3-ITS is very common in modern servers, on RPi4b only GICv2 is available.

virtio-mmio removed

v0.11.0

github-actions released this on Oct 30, 2020

v0.11.0

This release has been tracked through the 0.11.0 project.

Highlights for `cloud-hypervisor` version 0.11.0 include:

`io_uring` support by default for `virtio-block`

Provided that the host OS supports it (Linux kernel 5.8+) then `io_uring` will be used for a significantly higher performance block device.

Windows Guest Support

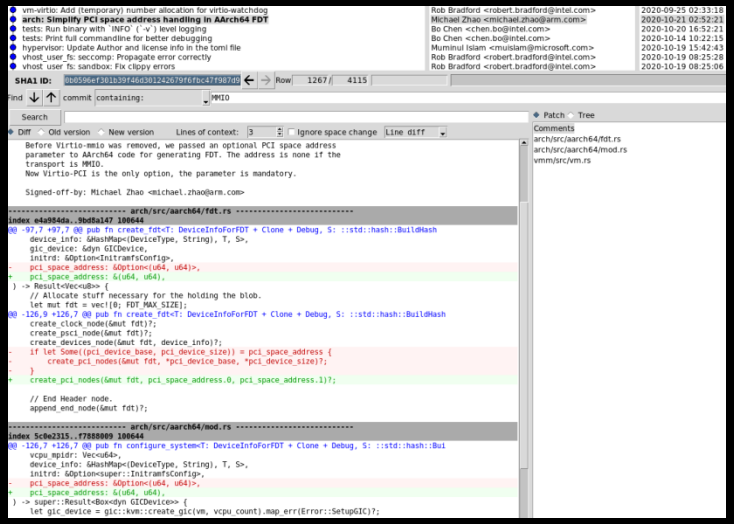
This is the first release where we officially support Windows running as a guest. Full details of how to setup the image and run Cloud Hypervisor with a Windows guest can be found in the dedicated Windows documentation.

`vhost-user` "Self Spawning" Deprecation

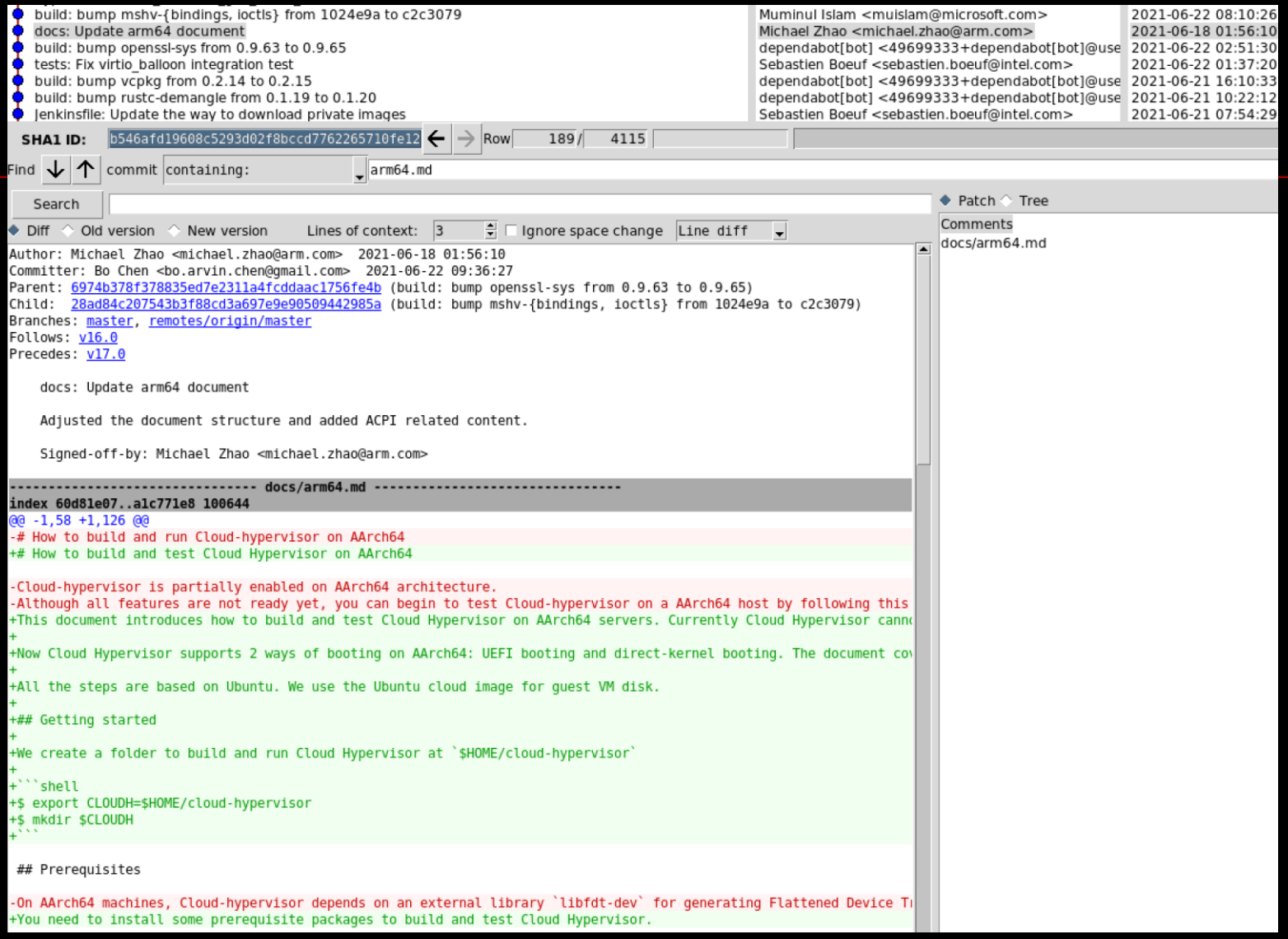
Automatically spawning a `vhost-user-net` or `vhost-user-block` backend is now deprecated. Users of this functionality will receive a warning and should make adjustments. The functionality will be removed in the next release.

`virtio-mmmio` Removal

Support for using the `virtio-mmio` transport, rather than using PCI, has been removed. This has been to simplify the code and significantly reduce the testing burden of the project.



document changes:



...

Firecracker supports GICv2

```
[mydev@fedora firecracker-main]$ tree -L 3 src/arch/src/aarch64/gic
src/arch/src/aarch64/gic
├── gicv2
│   ├── mod.rs
│   └── regs
│       ├── dist_regs.rs
│       ├── icc_regs.rs
│       └── mod.rs
├── gicv3
│   ├── mod.rs
│   └── regs
│       ├── dist_regs.rs
│       ├── icc_regs.rs
│       ├── mod.rs
│       └── redist_regs.rs
├── mod.rs
└── regs.rs
```

checking the code base…

Try to build the Guest Kernel

- https://github.com/cloud-hypervisor/cloud-hypervisor/blob/master/docs/arm64.md

```
[mydev@fedora linux-cloud-hypervisor-ch-5.12]$ cp ../cloud-hypervisor-master/resources/linux-config-aarch64 arch/arm64/configs/linux-config-aarch64_defconfig
[mydev@fedora linux-cloud-hypervisor-ch-5.12]$
[mydev@fedora linux-cloud-hypervisor-ch-5.12]$ git status
On branch ch-5.12
Your branch is up to date with 'origin/ch-5.12'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        arch/arm64/configs/linux-config-aarch64_defconfig

nothing added to commit but untracked files present (use "git add" to track)
[mydev@fedora linux-cloud-hypervisor-ch-5.12]$
```

failed:

```
arch/arm64/kernel/signal.c: In function ‘setup_return’:
./arch/arm64/include/asm/vdso.h:26:18: error: ‘vdso_offset_sigtramp’ undeclared (first use in this function)
   26 |         (void *)(vdso_offset_##name - VDSO_LBASE + (unsigned long)(base)); \
      |                  ^~~~~~~~~~~~
arch/arm64/kernel/signal.c:758:28: note: in expansion of macro ‘VDSO_SYMBOL’
  758 |                 sigtramp = VDSO_SYMBOL(current->mm->context.vdso, sigtramp);
      |                            ^~~~~~~~~~~
./arch/arm64/include/asm/vdso.h:26:18: note: each undeclared identifier is reported only once for each function it appears in
   26 |         (void *)(vdso_offset_##name - VDSO_LBASE + (unsigned long)(base)); \
      |                  ^~~~~~~~~~~~
arch/arm64/kernel/signal.c:758:28: note: in expansion of macro ‘VDSO_SYMBOL’
  758 |                 sigtramp = VDSO_SYMBOL(current->mm->context.vdso, sigtramp);
      |                            ^~~~~~~~~~~
make[2]: *** [scripts/Makefile.build:271: arch/arm64/kernel/signal.o] Error 1
make[1]: *** [scripts/Makefile.build:514: arch/arm64/kernel] Error 2
make[1]: *** Waiting for unfinished jobs....
```

patch1:

http://www.cxyzjd.com/article/u010164190/106166984

```
error: 'vdso_offset_sigtramp' undeclared (first use in this function)
  (void *)(vdso_offset_##name - VDSO_LBASE + (unsigned long)(base)); \
更改文件
更改arch/arm64/kernel/vdso/gen_vdso_offsets.sh脚本
中
's/^\([0-9a-fA-F]*\) . VDSO_\([a-zA-Z0-9_]*\)$/\#define vdso_offset_\2\t0x\1/p'
改为
's/^\([0-9a-fA-F]*\) . VDSO_\([a-zA-Z0-9_]*\)$/\#define vdso_offset_\2  0x\1/p'
```

but that in $KERNEL-CH-5.12_SRC/arch/arm64/kernel/vdso/ gen_vdso_offsets.sh has been changed.

patch2:

~~https://stackoverflow.com/questions/66369613/failed-to-build-arm64-kernel-to-halium~~

```
[mydev@fedora linux-cloud-hypervisor-ch-5.12]$ git status
On branch ch-5.12
Your branch is up to date with 'origin/ch-5.12'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   arch/arm64/Makefile

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        arch/arm64/configs/linux-config-aarch64_defconfig

no changes added to commit (use "git add" and/or "git commit -a")
[mydev@fedora linux-cloud-hypervisor-ch-5.12]$
[mydev@fedora linux-cloud-hypervisor-ch-5.12]$ git diff
diff --git a/arch/arm64/Makefile b/arch/arm64/Makefile
index 5b84aec31..d274a6f3f 100644
--- a/arch/arm64/Makefile
+++ b/arch/arm64/Makefile
@@ -169,6 +169,10 @@ Image.%: Image
 zinstall install:
        $(Q)$(MAKE) $(build)=$(boot) $@

+prepare: vdso_prepare
+vdso_prepare: prepare0
+        $(Q)$(MAKE) $(build)=arch/arm64/kernel/vdso include/generated/vdso-offsets.h
+
 PHONY += vdso_install
 vdso_install:
        $(Q)$(MAKE) $(build)=arch/arm64/kernel/vdso $@
[mydev@fedora linux-cloud-hypervisor-ch-5.12]$
```

but the same failure still occurred…

- **Investigation:**

  both arch/arm64/kernel/vdso/gen_vdso_offsets.sh and arch/arm64/Makefile are the same between https://github.com/cloud-hypervisor/linux.git(branch ch-5.12) & https://github.com/raspberrypi/linux(branch 5.12.y), while the latter does not have this vdso issue.

  seems an issue of my dev env…

- **Root cause:**

```
export PATH=$FGIT_HOME/bin:$JAVA_HOME/bin:$BAZEL_HOME:$GRADLE_HOME/bin:$GO_HOME/bin:$DOTNET_ROOT:$HOME/.dotnet/tools:$POWERSHELL_ROOT:$PATH:$HOME/.local/bin:$SUBLIME_TEXT_HOME:$GO_HOME/pkg/tool/linux_arm64
#export PATH=$FGIT_HOME/bin:$JAVA_HOME/bin:$BAZEL_HOME:$GRADLE_HOME/bin:$GO_HOME/bin:$GO_HOME/pkg/tool/linux_arm64:$DOTNET_ROOT:$HOME/.dotnet/tools:$POWERSHELL_ROOT:$PATH:$HOME/.local/bin:$SUBLIME_TEXT_HOME
```

  a mistake made by me in ~/.bashrc(the commented lined is where the issue happened, and the uncommented is the corrected version)

- **Successfully built the kernel(ch-5.12) without any patch, then rested it, and failed as expected(due to the known blocking issue of GICv2):**

```
[mydev@fedora linux-cloud-hypervisor-ch-5.12]$ ls -l arch/arm64/boot/Image
-rw-r--r-- 1 mydev mydev 20935168 Aug  8 01:12 arch/arm64/boot/Image
[mydev@fedora linux-cloud-hypervisor-ch-5.12]$
[mydev@fedora linux-cloud-hypervisor-ch-5.12]$ cp arch/arm64/boot/Image /opt/MyWorkSpace/Tmp/Virtual/Cloud-Hypervisor/kernel-ch-5.12.bin
[mydev@fedora linux-cloud-hypervisor-ch-5.12]$
```

```
[mydev@fedora Cloud-Hypervisor]$ ll
total 2122488
drwxr-xr-x. 1 mydev mydev        370 Aug 10 00:47 ./
drwxr-xr-x. 1 mydev mydev         80 Aug  7 04:19 ../
-rwxr-xr-x. 1 mydev mydev   62331224 Aug  7 04:20 cloud-hypervisor*
-rw-r--r--. 1 mydev mydev  527433728 Aug  3 15:24 focal-server-cloudimg-arm64.img
-rw-r--r--. 1 mydev mydev 2361393152 Aug  7 04:50 focal-server-cloudimg-arm64.raw
-rw-r--r--. 1 mydev mydev      85928 Jul 23 01:42 hypervisor-fw
-rw-r--r--. 1 mydev mydev    8356352 Nov 24  2020 kernel.bin
-rw-r--r--  1 mydev mydev   20935168 Aug 10 00:47 kernel-ch-5.12.bin
-rw-------. 1 root  root        4332 Aug  7 04:53 log-capset.log
-rw-------. 1 root  root        4314 Aug  7 04:35 log.log
-rw-------. 1 root  root           0 Aug  7 04:53 serial-capset.log
-rw-------. 1 root  root           0 Aug  7 04:35 serial.log
-rw-r--r--. 1 mydev mydev  209715200 Sep 10  2019 xenial.rootfs.ext4
[mydev@fedora Cloud-Hypervisor]$
[mydev@fedora Cloud-Hypervisor]$ sudo ./cloud-hypervisor --api-socket /tmp/cloud-hypervisor.sock --kernel ./kernel-ch-5.12.bin --disk path=focal-server-cloudimg-arm64.raw --cmdline "keep_bootcon console=ttyAMA0 reboot=k panic=1 root=/dev/vda1 rw" --cpus boot=4 --memory size=4096M --serial tty --console off --log-file log.log -vvv --net "tap=,mac=,ip=,mask="
[sudo] password for mydev:
Error booting VM: VmBoot(ConfigureSystem(AArch64Setup(SetupGic(CreateGic(CreateDevice(No such device (os error 19)))))))
[mydev@fedora Cloud-Hypervisor]$
```

1.2 Summary

No need to install rustup

- For directly build on RPi4

```
# Install rust tool chain
$ curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

Currently Cloud-Hypervisor does not support RPi4

- The blocking GICv2 issue
  working on re-add GICv2 support in Cloud-Hypervisor…

Carefully read the guide before taking any action

- Firstly, ensure it is for ARM!
  …

2) Kata Containers on RPi4

Setup

- sudo dnf install libvirt qemu-kvm oci-kvm-hook virt-install genisoimage libvirt-client libvirt-python3 libvirt-daemon-kvm libvirt-daemon-config-network fakeroot policycoreutils-python-utils libguestfs qemu-user-static kpartx jq podman skopeo buildah wine…
- https://docs.docker.com/engine/install/fedora/
-

```
[mydev@fedora ~]$ which kata-runtime
/usr/bin/kata-runtime
[mydev@fedora ~]$
[mydev@fedora ~]$ kata-runtime check
/usr/share/kata-containers/defaults/configuration.toml: file /usr/bin/qemu-kvm does not exist
[mydev@fedora ~]$
[mydev@fedora ~]$ cd /usr/bin
[mydev@fedora bin]$ sudo ln -sf qemu-system-aarch64 qemu-kvm
[mydev@fedora bin]$
[mydev@fedora bin]$ ls -l qemu-kvm
lrwxrwxrwx. 1 root root 19 May 19 02:15 qemu-kvm -> qemu-system-aarch64
[mydev@fedora bin]$
[mydev@fedora bin]$ kata-runtime check
Invalid command "check"
[mydev@fedora bin]$ kata-runtime kata-check
Newer minor release available: 2.1.0 (url: https://github.com/kata-containers/kata-containers/releases/download/2.1.0/kata-static-2.1.0-x86_64.tar.xz,
 date: 2021-05-14T18:37:59Z)
WARN[0001] modprobe insert module failed: modprobe: ERROR: could not insert 'vhost_net': Operation not permitted  arch=arm64 error="exit status 1" mod
ule=vhost_net name= pid=19893 source=runtime
ERRO[0001] kernel property not found                     arch=arm64 description="Host kernel accelerator for virtio network" name=vhost_net pid=19893
source=runtime type=module
WARN[0001] modprobe insert module failed: modprobe: ERROR: could not insert 'vhost_vsock': Operation not permitted  arch=arm64 error="exit status 1" m
odule=vhost_vsock name= pid=19893 source=runtime
ERRO[0001] kernel property not found                     arch=arm64 description="Host Support for Linux VM Sockets" name=vhost_vsock pid=19893 source=
runtime type=module
WARN[0001] modprobe insert module failed: modprobe: ERROR: could not insert 'vhost': Operation not permitted  arch=arm64 error="exit status 1" module=
vhost name= pid=19893 source=runtime
ERRO[0001] kernel property not found                     arch=arm64 description="Host kernel accelerator for virtio" name=vhost pid=19893 source=runti
me type=module
ERRO[0001] ERROR: System is not capable of running Kata Containers  arch=arm64 name= pid=19893 source=runtime
ERROR: System is not capable of running Kata Containers
[mydev@fedora bin]$
[mydev@fedora bin]$ sudo kata-runtime kata-check
WARN[0000] Not running network checks as super user      arch=arm64 name= pid=20048 source=runtime
System is capable of running Kata Containers
System can currently create Kata Containers
[mydev@fedora bin]$
```

Kata Env

```
[mydev@fedora /]$ kata-runtime kata-env
[Meta]
  Version = "1.0.25"

[Runtime]
  Debug = false
  Trace = false
  DisableGuestSeccomp = true
  DisableNewNetNs = false
  SandboxCgroupOnly = true
  Path = "/usr/bin/kata-runtime"
  [Runtime.Version]
    OCI = "1.0.1-dev"
    [Runtime.Version.Version]
      Semver = "2.1.0"
      Major = 2
      Minor = 1
      Patch = 0
      Commit = "04c0183bceafe68bae317809de6aad0d47679dd1"
  [Runtime.Config]
    Path = "/usr/share/kata-containers/defaults/configuration.toml"

[Hypervisor]
  MachineType = "virt"
  Version = "QEMU emulator version 5.2.0 (qemu-5.2.0-8.fc34)\nCopyright (c) 2003-2020 Fabrice Bellard and the QEMU Project developers"
  Path = "/usr/bin/qemu-system-aarch64"
  BlockDeviceDriver = "virtio-scsi"
  EntropySource = "/dev/urandom"
  SharedFS = "virtio-fs"
  VirtioFSDaemon = "/usr/libexec/virtiofsd"
  Msize9p = 8192
  MemorySlots = 10
  PCIeRootPort = 0
  HotplugVFIOOnRootBus = false
  Debug = false

[Image]
  Path = ""

[Kernel]
  Path = "/usr/lib/modules/5.13.7-200.fc34.aarch64/vmlinuz"
  Parameters = "scsi_mod.scan=none"

[Initrd]
  Path = "/var/cache/kata-containers/osbuilder-images/5.13.7-200.fc34.aarch64/fedora-kata-5.13.7-200.fc34.aarch64.initrd"

[Agent]
  Debug = false
  Trace = false
  TraceMode = ""
  TraceType = ""
```

Blocking issue

- **https://bugzilla.redhat.com/show_bug.cgi?id=1912162 //Starting podman container with kata-runtime fails with SELinux errors said to be no longer reproduced on Fedora 34, but…**

```
[mydev@fedora /]$ /usr/sbin/sestatus -v
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33

Process contexts:
Current context:                unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
Init context:                   system_u:system_r:init_t:s0

File contexts:
Controlling terminal:           unconfined_u:object_r:user_devpts_t:s0
/etc/passwd                     system_u:object_r:passwd_file_t:s0
/etc/shadow                     system_u:object_r:shadow_t:s0
/bin/bash                       system_u:object_r:shell_exec_t:s0
/bin/login                      system_u:object_r:login_exec_t:s0
/bin/sh                         system_u:object_r:bin_t:s0 -> system_u:object_r:shell_exec_t:s0
/sbin/agetty                    system_u:object_r:getty_exec_t:s0
/sbin/init                      system_u:object_r:bin_t:s0 -> system_u:object_r:init_exec_t:s0
/usr/sbin/sshd                  system_u:object_r:sshd_exec_t:s0
[mydev@fedora /]$
```

then disabled SELinux and retried:

```
[mydev@fedora Kata-Containers]$ podman images
REPOSITORY                      TAG         IMAGE ID      CREATED       SIZE
docker.io/library/alpine        latest      ae607a46d002  33 hours ago  5.61 MB
registry.access.redhat.com/ubi8 latest      795e28a8b49a  2 weeks ago   267 MB
docker.io/library/busybox       latest      d9d6c2bcb750  2 months ago  1.63 MB
[mydev@fedora Kata-Containers]$
```

```
[mydev@fedora /]$ sestatus -v
SELinux status:                 disabled
[mydev@fedora /]$
[mydev@fedora /]$
[mydev@fedora /]$ podman run -it --rm --runtime=/usr/bin/kata-runtime ubi8 bash
Invalid command "delete"
                        ERRO[0002] Error removing container 7cac917b441d4ff137fc8518dfc07202ad4c7053c4a5821f6ed8ad70a6f60876 from runtime after creation failed
Error: OCI runtime error: Invalid command "create"
[mydev@fedora /]$
[mydev@fedora /]$ podman run -it --rm --security-opt label=disable --runtime=/usr/bin/kata-runtime ubi8 bash
Invalid command "delete"
                        ERRO[0002] Error removing container d604947a55713d781d907ba7393e394b4fa7b2d2c1dedf98c4852e4fdd041a7d from runtime after creation failed
Error: OCI runtime error: Invalid command "create"
[mydev@fedora /]$
```

…

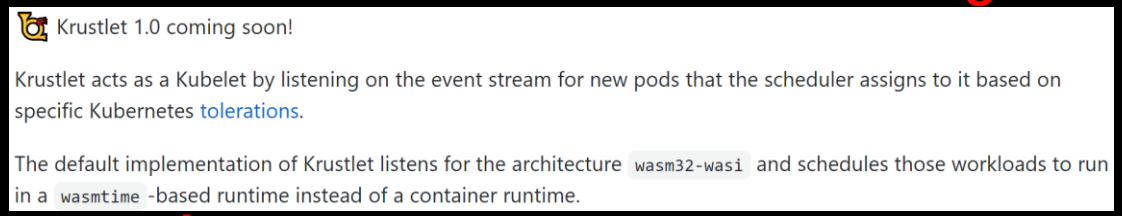
the same failures can also be reproduced on X64.

III. Rust for Cloud Native

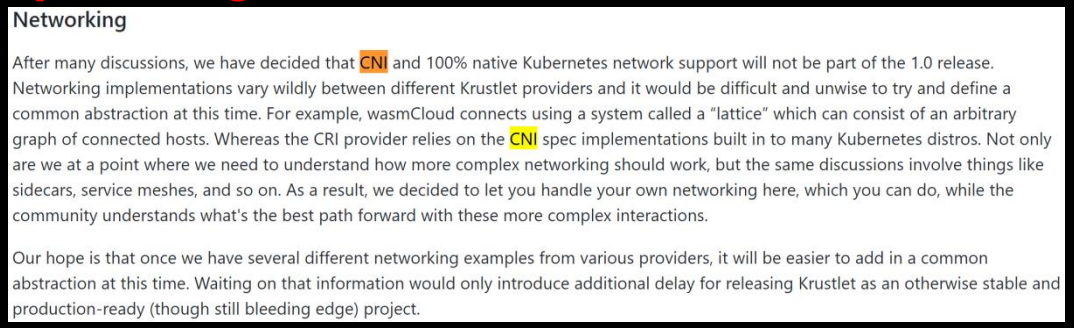
1) Rust for K8S

1.1 Krustlet

- https://krustlet.dev/
- https://github.com/krustlet/krustlet
- Kubernetes Kubelet in Rust for running WASM



- Upcoming v1.0



- https://cloudblogs.microsoft.com/opensource/2020/04/07/announcing-krustlet-kubernetes-rust-kubelet-webassembly-wasm/

Build from source on RPi4

- **https://docs.krustlet.dev/community/developers/**
  Krustlet officially using just(https://github.com/casey/just) as the build system, but you also use cargo:
  on Fedora: sudo dnf install just

```
[mydev@fedora krustlet-main]$ which just
/usr/bin/just
[mydev@fedora krustlet-main]$ just -V
just 0.9.8
[mydev@fedora krustlet-main]$
```

- **Using "tuna" as the Cargo repo:**

```
[mydev@fedora krustlet-main]$ just build
cargo build
    Updating `https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git` index
  Downloaded ansi_term v0.11.0 (registry `https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git`)
  Downloaded cap-fs-ext v0.13.10 (registry `https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git`)
  Downloaded base64 v0.10.1 (registry `https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git`)
  Downloaded cpufeatures v0.1.5 (registry `https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git`)
```

...

```
  Downloaded ring v0.16.20 (registry `https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git`)
  Downloaded k8s-openapi v0.12.0 (registry `https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git`)
  Downloaded prost-build v0.7.0 (registry `https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git`)
  Downloaded 324 crates (38.0 MB) in 6m 02s (largest was `prost-build` at 8.2 MB)
   Compiling libc v0.2.98
   Compiling proc-macro2 v1.0.27
   Compiling unicode-xid v0.2.2
   Compiling syn v1.0.74
   Compiling cfg-if v1.0.0
   Compiling autocfg v1.0.1
error: linker `aarch64-linux-gnu-gcc` not found
  |
  = note: No such file or directory (os error 2)

error: aborting due to previous error

error: could not compile `syn`

To learn more, run the command again with --verbose.
warning: build failed, waiting for other jobs to finish...
error: linker `aarch64-linux-gnu-gcc` not found
  |
  = note: No such file or directory (os error 2)

error: aborting due to previous error

error: linker `aarch64-linux-gnu-gcc` not found
  |
  = note: No such file or directory (os error 2)

error: aborting due to previous error

error: build failed
error: Recipe `build` failed on line 12 with exit code 101
[mydev@fedora krustlet-main]$
```

patched as below, since this is in-device development, and rebuilt it again:

```
[mydev@fedora krustlet-main]$ git diff
diff --git a/.cargo/config b/.cargo/config
index e2f942b..9a01ee8 100644
--- a/.cargo/config
+++ b/.cargo/config
@@ -1,5 +1,5 @@
 [target.armv7-unknown-linux-gnueabihf]
-linker = "arm-linux-gnueabihf-gcc"
+linker = "gcc"

 [target.aarch64-unknown-linux-gnu]
-linker = "aarch64-linux-gnu-gcc"
+linker = "gcc"
[mydev@fedora krustlet-main]$
[mydev@fedora krustlet-main]$ just build
cargo build
   Compiling libc v0.2.98
   Compiling proc-macro2 v1.0.27
   Compiling unicode-xid v0.2.2
   Compiling syn v1.0.74
```

...

```
   Compiling cranelift-codegen v0.75.0
   Compiling k8s-csi v0.3.0
   Compiling kubelet v1.0.0-alpha.1 (/opt/MyWorkSpace/MyProjs/HCI/Kubernetes/WASMonK8S/Deislabs/krustlet-main/crates/kubelet)
error: failed to run custom build command for `k8s-csi v0.3.0`

Caused by:
  process didn't exit successfully: `/opt/MyWorkSpace/MyProjs/HCI/Kubernetes/WASMonK8S/Deislabs/krustlet-main/target/debug/build/k8s-csi-06ad4823123cb3d9/build-script-build` (exit status: 1)
  --- stderr
  error running rustfmt: Os { code: 2, kind: NotFound, message: "No such file or directory" }
warning: build failed, waiting for other jobs to finish...
error: build failed
error: Recipe `build` failed on line 12 with exit code 101
[mydev@fedora krustlet-main]$
```

"no rustfmt" error, just install it via "sudo dnf install rustfmt", and rebuilt:

...

```
Compiling oci-distribution v0.7.0 (/opt/MyWorkSpace/MyProjs/HCI/Kubernetes/WASMonK8S/Deislabs/krustlet-main/crates/oci-distribution)
Compiling kube-core v0.58.1
Compiling kube v0.58.1
Compiling cranelift-frontend v0.75.0
Compiling cranelift-native v0.75.0
Compiling cranelift-wasm v0.75.0
Compiling kube-runtime v0.58.1
Compiling wasmtime-environ v0.28.0
Compiling krator v0.4.0
Compiling wasmtime-debug v0.28.0
Compiling wasmtime-cranelift v0.28.0
Compiling wasmtime-obj v0.28.0
Compiling wasmtime-profiling v0.28.0
Compiling wasmtime-jit v0.28.0
Compiling zstd v0.6.1+zstd.1.4.9
Compiling wasmtime v0.28.0
Compiling wiggle v0.28.0
Compiling wasi-cap-std-sync v0.28.0
Compiling wasi-experimental-http-wasmtime v0.5.0
Compiling wasi-provider v1.0.0-alpha.1 (/opt/MyWorkSpace/MyProjs/HCI/Kubernetes/WASMonK8S/Deislabs/krustlet-main/crates/wasi-provider)
Compiling krustlet v1.0.0-alpha.1 (/opt/MyWorkSpace/MyProjs/HCI/Kubernetes/WASMonK8S/Deislabs/krustlet-main)
 Finished dev [unoptimized + debuginfo] target(s) in 21m 58s
```

though Krustlet has no official release for AArch64 by now, but it has built-in support for ARM actually, so we can successfully built it on RPi4:

```
[mydev@fedora krustlet-main]$ tree -L 1 target/debug
target/debug
├── build
├── deps
├── examples
├── incremental
├── krustlet-wasi
├── krustlet-wasi.d
├── oneclick
├── oneclick.d
├── podsmiter
└── podsmiter.d

4 directories, 6 files
[mydev@fedora krustlet-main]$ ls -l target/debug/krustlet-wasi
-rwxr-xr-x 2 mydev mydev 425407168 Aug 10 02:27 target/debug/krustlet-wasi
[mydev@fedora krustlet-main]$ file target/debug/krustlet-wasi
target/debug/krustlet-wasi: ELF 64-bit LSB pie executable, ARM aarch64, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-aarch64.so.1, BuildID[sha1]=b33ee4bdcddc260e108db3479cf6c1c3ea
06d02e, for GNU/Linux 3.7.0, with debug_info, not stripped
[mydev@fedora krustlet-main]$
```

...

Krustlet for lightweight K8S

- **https://thenewstack.io/krustlet-brings-webassembly-to-kubernetes-with-a-rust-based-kubelet/**

> "Krustlet, potentially combined with things like SUSE/Rancher's k3s, can make inroads into IoT by providing a small-footprint extension to a Kubernetes cluster. This points to a sea change occurring in Kubernetes. When some folks

- **https://docs.krustlet.dev/howto/krustlet-on-microk8s/**

snap install microk8s --classic

2) Good Resources

- https://www.cncf.io/blog/2020/06/22/rust-at-cncf/
- https://events.linuxfoundation.org/cloud-native-rust-day/
- https://buoyant.io/media/why-the-future-of-the-cloud-will-be-built-on-rust/
- https://github.com/rust-cloud-native
- …

IV. Wrap-up

- Open source projects for Cloud Infrastructure or Cloud Native that written in Rust are booming: Rust-VMM, Firecracker, Cloud-Hypervisor, Krustlet… and those for Linux Kernel.

- In general, some open source projects are not fully tested on ARM, and developer may meet different issues across various distributions.

- A new war of system programming language is only just beginning…

Q & A

Thanks!

Reference

Slides/materials from many and varied sources:

- http://en.wikipedia.org/wiki/
- http://www.slideshare.net/
- https://en.wikipedia.org/wiki/Comparison_of_application_virtualization_software
- https://www.sciencedirect.com/topics/computer-science/assisted-virtualization
- https://en.wikipedia.org/wiki/Systems_programming
- https://docs.01.org/clearlinux/latest/tutorials/kata.html
- https://www.redhat.com/sysadmin/selinux-kata-containers
- https://searchservervirtualization.techtarget.com/feature/Virtualization-trends-focus-on-HCI-Kubernetes