Add RISC-V support to your favorite Operating System

RISC-V Devroom, FOSDEM'25
February 1, 2025

About

- Adrian Vladu
- Flatcar maintainer
- Cloud Engineer
- Github: @ader1990
- Email: avladu@cloudbasesolutions.com

Outline

- Why RISC-V
- Why Flatcar Container Linux
- Adding RISC-V support to Flatcar Container Linux
- Demo time: Flatcar running on RISC-V

History

- Found out about RISC-V two years ago at FOSDEM
- Last year I saw some RISC-V boards at FOSDEM
- Decided to buy one RISC-V board and see how it goes
- Been tinkering for the past 10 years with ARM64 boards

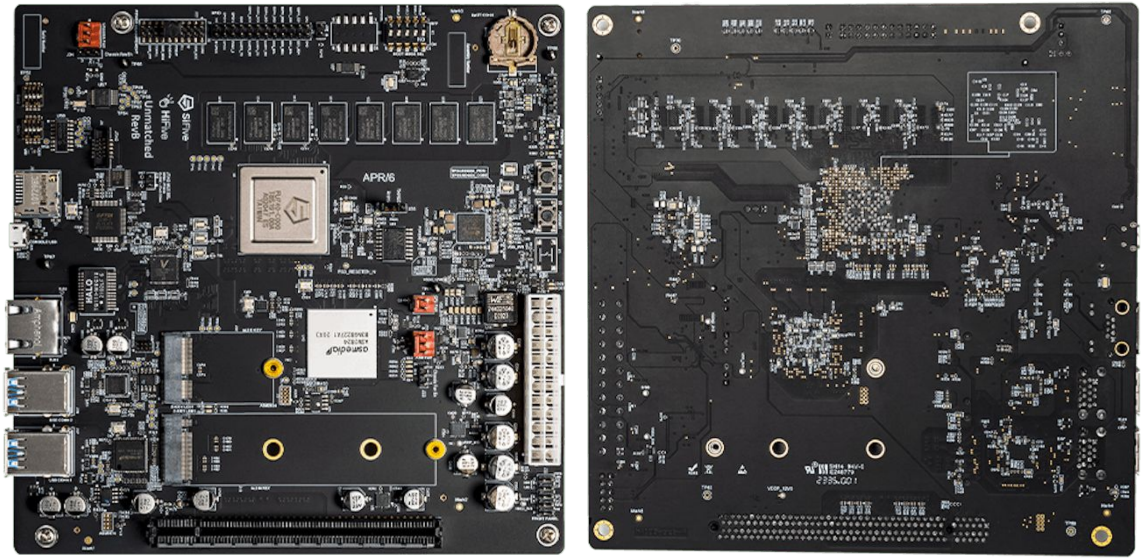
Owned RISC-V Boards

LicheePi4A

Owned RISC-V Boards

HiFive

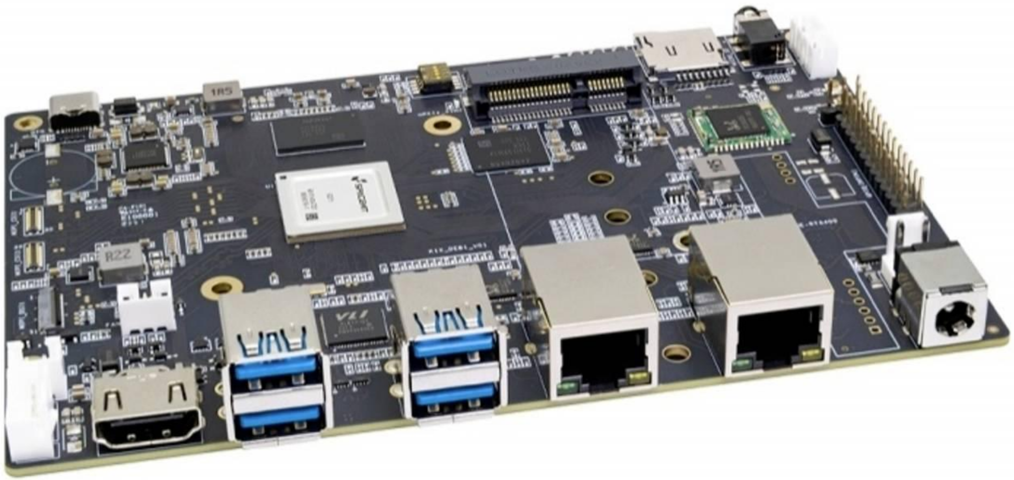
Unmatched



Owned RISC-V Boards

Banana Pi

F3



Why RISC-V

- FOSS ISA - ‘Linux equivalent of the CPU ISAs’
- Recent uptick in hardware availability
  - Personal ownership
  - Cloud based pay-per-time
- For fun / tinkering with something new
- Bright future ahead

Why Flatcar Container Linux

- Compact, easy to run
- Image based operating system
- Secure and reliable by design
- Containerization, kubernetization, Cloud Native world

Add RISC-V support: Flatcar ancestry

- Drop-in replacement for CoreOS
- Downstream fork of Gentoo
  - https://fosdem.org/2025/schedule/event/fosdem-2025-4763-flatcar-and-gentoo-sitting-in-a-tree-a-collaboration-of-distributions/
  - James "Chewi" Le Cuirot, Sunday, 9:30 AM, H.1302 (Depage).
- Immutable OS
- Upgrades done via A/B partitioning scheme

Add RISC-V support: Gentoo

- Gentoo already has support for RISC-V (unstable ~riscv)
- Investigated if I could run Gentoo on my LicheePi4A
- Had to borrow the Linux kernel from the LicheePI4A
- Gentoo worked afterwards
- Every package needs to be compiled (very slow)
- Flatcar should solve this issue via Docker and Alpine community

Add RISC-V support: Flatcar bill of materials

- Flatcar SDK (Dockerized)
- Flatcar packages
- Systemd-sysexts
- Flatcar image:
  - Bootloader
  - dm-verity
  - Virtualization

Add RISC-V support: Flatcar SDK

- Docker image
- Based on Gentoo’s bootstrap workflow
- Flatcar currently supports AMD64 and ARM64
- ARM64 is cross-compiled on AMD64 machines
- Use same workflow for RISC-V

Add RISC-V support: Flatcar SDK

- riscv64-cros-linux-gnu
- riscv64gc-unknown-linux-gnu
- Add profile based on the Gentoo ~riscv profile
  - Inherit default/linux/riscv/23.0/rv64/lp64d/systemd
  - GCC flags: march=rv64gc -mabi=lp64d -g
- Install rustc cross-compiler for RISC-V (Flatcar requires ~500 crates)
- Install qemu-riscv64-static / binfmt
  - Needed by some packages Makefile

Add RISC-V support: Flatcar packages

- coreos-overlay packages
    - Inherited from CoreOS
    - Some are Flatcar only packages
    - Updated manually and infrequently
- Stable packages
    - Gentoo
    - Mirrored once a week
- Most of the work needed was on the coreos-overlay packages
    - To add the ~riscv keyword
- Some stable Gentoo packages needed work too
    - Work was upstreamed on some already

Add RISC-V support: Flatcar packages

- Boilerplate work done
- Success building the Flatcar SDK?

Add RISC-V support: Flatcar packages

- Limit packages to only AMD64/ARM64 architectures
  - kexec-tools
  - azure-vm-utils
  - app-emulation/amazon-ssm-agent
  - sys-boot/shim
  - sys-boot/shim-signed
- Enable/fix upstream Gentoo support for:
  - tpm2-tools
  - efivar
  - Python libffi: ffi.h not found
  - Golang architecture mapping
- Map correctly the target for RUSTC
  - RISCV:riscv64gc-unknown-linux-gnu:riscv64-cros-linux-gnu

Add RISC-V support: Flatcar packages

- Linux Kernel
  - Remove Xen and Hyper-V from the RISC-V config
  - Testing with QEMU software emulation required extra work
  - CONFIG_USB_XHCI_PLATFORM=y
  - CONFIG_PCI_HOST_GENERIC=y
  - efi=debug earlycon=sbi
  - add risc-v qemu_fw_cfg support to Linux kernel 6.6
- Docker
  - Needed some build change related to LD_PRELOAD

Add RISC-V support: Flatcar systemd-sysext

- Enable RISCV64 architecture for sysexts
- Docker
- Kubernetes
- Now we have a ready Flatcar root filesystem built with all the packages
- Success?

Add RISC-V support: Flatcar images

- Bootloader
  - Use RHEL patches on top of grub2
  - Fix Flatcar grub2 patches for RISC-V (inherited from CoreOS)
  - Investigate/Document OpenSBI or EDK2 to boot Flatcar on QEMU software emulation
- dm-verity
  - Add 0x160 (352) offset where to put the verity hash in the vmlinuz
- Virtualization
  - Unfortunately, I have tried to enable CONFIG_KVM=y on my Banana Pi F3 with no actual success. There “seems” to be no available RISC-V CPU on the market, with H extension (virtualization extension)
  - The only option now is to use QEMU software emulation
  - https://www.linkedin.com/posts/robin-randhawa-8335423_bhyve-hypervisor-on-hifive-premier-p550-activity-7273022906392027137-AEMc

Add RISC-V support: Flatcar demo time

- QEMU emulation
- EDK2
- OpenSBI
- https://github.com/ader1990/scripts/releases/tag/riscv-poc-07-jan-2025

Add RISC-V support: Flatcar demo time

- Docker and containerd come as systemd-sysexts
  - Compilation not needed anymore
- K3S is a viable option on RISC-V to run Kubernetes
  - https://github.com/CARV-ICS-FORTH/kubernetes-riscv64?tab=readme-ov-file#acknowledgements

Add RISC-V support: Flatcar demo time

QEMU boot using edk2 tianocore

```
IMAGE_PATH="flatcar_production_qemu_uefi_image.img"

# RISCV_VIRT_CODE and RISCV_VIRT_VARS downloadable from https://dev.gentoo.org/~chewi/distfiles/edk2-202411-1-riscv.x

qemu-system-riscv64 \
-machine virt,acpi=off -cpu rv64 \
-m 8192 -smp 16 \
-drive if=pflash,format=qcow2,unit=0,file=RISCV_VIRT_CODE.qcow2,readonly=on \
-drive if=pflash,format=qcow2,unit=1,file=RISCV_VIRT_VARS.qcow2 \
-device virtio-net-device,netdev=eth0 -netdev user,id=eth0 \
-device virtio-rng-pci \
-drive file=$IMAGE,format=raw,if=virtio \
-nographic -vnc :1 \
-serial mon:stdio \
-device virtio-gpu-pci \
-device qemu-xhci,id=xhci -device usb-kbd,bus=xhci.0

# Enter Boot Manager and boot from file
```

Add RISC-V support: Flatcar demo time

```
localhost ~ # cat /etc/os-release
NAME="Flatcar Container Linux by Kinvolk"
ID=flatcar
ID_LIKE=coreos
VERSION=4187.0.0+nightly-20241217-2100
VERSION_ID=4187.0.0
BUILD_ID=nightly-20241217-2100
SYSEXT_LEVEL=1.0
PRETTY_NAME="Flatcar Container Linux by Kinvolk 4187.0.0+nightly-20241217-2100 (Oklo)"
ANSI_COLOR="38;5;75"
HOME_URL="https://flatcar.org/"
BUG_REPORT_URL="https://issues.flatcar.org"
FLATCAR_BOARD="riscv-usr"
CPE_NAME="cpe:2.3:o:flatcar-linux:flatcar_linux:4187.0.0+nightly-20241217-2100:*:*:*:*:*:*"
localhost ~ # uname -a
Linux localhost 6.6.65-flatcar #1 SMP Wed Jan  1 21:44:12 -00 2025 riscv64 GNU/Linux
localhost ~ # systemctl --version
systemd 255 (255)
+PAM +AUDIT +SELINUX -APPARMOR +IMA +SMACK +SECCOMP +GCRYPT -GNUTLS +OPENSSL -ACL +BLKID +CURL
+PCRE2 -PWQUALITY -P11KIT -QRENCODE +TPM2 +BZIP2 +LZ4 +XZ +ZLIB +ZSTD -BPF_FRAMEWORK -XKBCOMMON
```

Add RISC-V support: Flatcar demo time

```
localhost ~ # systemd-sysext list
NAME              TYPE PATH                                     TIME
containerd-flatcar raw  /etc/extensions/containerd-flatcar.raw Tue 2025-01-07 10:23:54 UTC
docker-flatcar     raw  /etc/extensions/docker-flatcar.raw     Tue 2025-01-07 10:24:37 UTC
k3s                raw  /etc/extensions/k3s.raw                Wed 2025-01-08 07:29:31 UTC
oem-qemu           raw  /etc/extensions/oem-qemu.raw           Tue 2025-01-07 10:38:58 UTC
localhost ~ # docker --version
Docker version 27.3.1, build ce1223035ac3ab8922717092e63a184cf67b493d
localhost ~ # containerd --version
containerd github.com/containerd/containerd v1.7.23 57f17b0a6295a39009d861b89e3b3b87b005ca27
localhost ~ # kubectl version
Client Version: v1.31.1+k3s-34f8fa8d
Kustomize Version: v5.4.2
Server Version: v1.31.1+k3s-34f8fa8d
```

Add RISC-V support: Flatcar demo time

```
localhost ~ # docker run hello-world
[44715.964962] docker0: port 1(veth715fc54) entered blocking state
[44715.969050] docker0: port 1(veth715fc54) entered disabled state
[44715.970191] veth715fc54: entered allmulticast mode
[44715.977872] veth715fc54: entered promiscuous mode
[44723.507439] eth0: renamed from vethfd731d0
[44723.548313] docker0: port 1(veth715fc54) entered blocking state
[44723.548843] docker0: port 1(veth715fc54) entered forwarding state

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (riscv64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/
```

Add RISC-V support: Flatcar demo time

```
localhost ~ # kubectl get node -A
kubectl get pods -A
NAME        STATUS   ROLES                  AGE   VERSION
localhost   Ready    control-plane,master   24d   v1.31.1+k3s-34f8fa8d
localhost ~ # kubectl get pods -A
NAMESPACE     NAME                                      READY   STATUS      RESTARTS
default       hello-5c6fbd75ff-tpqn8                    1/1     Running     2 (12h ago)
kube-system   coredns-5f499f5dcb-8dpkp                  1/1     Running     4 (12h ago)
kube-system   helm-install-traefik-9v2hp                0/1     Completed   7
kube-system   helm-install-traefik-crd-t45fh            0/1     Completed   0
kube-system   local-path-provisioner-ff964b654-kdg7s    1/1     Running     3 (12h ago)
kube-system   metrics-server-56f4447f89-xrk5r           1/1     Running     4 (12h ago)
kube-system   svclb-traefik-c86a8609-fkxxm              2/2     Running     4 (12h ago)
kube-system   traefik-646c7c9654-hlnhx                  1/1     Running     4 (12h ago)
```

Add RISC-V support: Flatcar demo time

```
localhost ~ # cat get_verity_hash.sh
hexdump /boot/flatcar/vmlinuz-a | head -n 30 | grep -i 160 -A 4 -B 100
localhost ~ # bash get_verity_hash.sh
0000000 5a4d 106f 0ce0 0001 0000 0020 0000 0000
0000010 5000 049f 0000 0000 0000 0000 0000 0000
0000020 0002 0000 0000 0000 0000 0000 0000 0000
0000030 4952 4353 0056 0000 5352 0543 0040 0000
0000040 4550 0000 5064 0002 0000 0000 0000 0000
0000050 0000 0000 00a0 0206 020b 1402 f000 009f
0000060 5000 03ff 0000 0000 f604 0083 1000 0000
0000070 0000 0000 0000 0000 1000 0000 0200 0000
0000080 0000 0000 0003 0000 0000 0000 0000 0000
0000090 5000 049f 1000 0000 0000 0000 000a 0100
00000a0 0000 0000 0000 0000 0000 0000 0000 0000
*
00000c0 0000 0000 0006 0000 0000 0000 0000 0000
00000d0 0000 0000 0000 0000 0000 0000 0000 0000
*
00000f0 0000 0000 0000 0000 742e 7865 0074 0000
0000100 f000 009f 1000 0000 f000 009f 1000 0000
0000110 0000 0000 0000 0000 0000 0000 0020 6000
0000120 642e 7461 0061 0000 5000 03ff 0000 00a0
0000130 be00 03f4 0000 00a0 0000 0000 0000 0000
0000140 0000 0000 0040 c000 0013 0000 0013 0000
0000150 0013 0000 0013 0000 0013 0000 0013 0000
0000160 3637 3934 3036 3862 3539 3336 6332 3561
```

Add RISC-V support: Flatcar demo time

```
localhost ~ # cat cmdline.txt
BOOT_IMAGE=/flatcar/vmlinuz-a mount.usr=/dev/mapper/usr
 verity.usr=PARTUUID=7130c94a-213a-4e5a-8e26-6cce9662f132
rootflags=rw mount.usrflags=ro consoleblank=0 root=LABEL=ROOT
 efi=debug earlycon=sbi flatcar.oem.id=qemu flatcar.autologin
 verity.usrhash=764960b895632ca5b7b2fbf41ba3ac33af41fb2261ab1bd8d44a253c39c6643
```

Add RISC-V support: Takeaways

- Software support looks great on RISC-V
- Most of the issues that had to be solved were related to cross-compiling and mix-and-match arch quirks
- Kernel/bootloader/dm-verity required special attention
- One can run Kubernetes on RISC-V, which opens the door to <ANY> software
- Testing is slow as the only options now are
  - Test on hardware boards - takes time to flash SDcards/NVMEs and build special kernel sources, special kernel CONFIGs (rabbit hole)
  - For example, because of this, Ubuntu has to ship a separate image for every board version
  - QEMU without KVM is very slow

Add RISC-V support: Flatcar future work

- More work required for official support
  - Feedback from community needed
  - CI integration
  - More polishing
- Run Flatcar on physical RISC-V boards
- Run Flatcar on QEMU KVM RISC-V vms
- Upstream all the existing patches to Gentoo

Add RISC-V support: Links

- Main Flatcar issue
  - https://github.com/flatcar/Flatcar/issues/1420
- Flatcar Work in progress:
  - https://github.com/ader1990/scripts/tree/ader1990/riscv-poc-v2
- Flatcar images available (currently not officially supported):
  - https://github.com/ader1990/scripts/releases/tag/riscv-poc-07-jan-2025

Add RISC-V support: Question time

- Ask me anything

Wait, but give me a real world scenario

- https://app.devcon.org/schedule/J3SWYT
- Ethereum is one of the biggest running software in the world
  - Biggest peer-to-peer Virtual Machine implementation by far
  - Multiple instances that converge and communicate
- Can latest gen RISC-V boards run it?
- Improvement needed to have the hashing algos implemented with RISC-V vector extensions
- Discussion in progress to have the Ethereum Virtual Machine (Stack based) implemented directly with RISC-V instruction set
- https://vitalik.eth.limo/general/2024/10/26/futures5.html