ORIGIN ACADEMY · COURSE SYLLABUS

Virtualization Engineer (QEMU & KVM)

Build and run virtual machines from the hypervisor up: QEMU, KVM, virtio, libvirt, live migration and writing your own emulated devices.

20 weeks · about 12 hours a week · live cohort of up to 10
Curriculum last updated 18 November 2026

Availability

This course opens soon. Join the waitlist on the course page to hear first about dates and prices.

Before you start

- Linux administration skills (our Linux Administration course)
- C programming, for the KVM and device-development phases
- English good enough to follow technical documentation and live sessions: about a credit (C6 or better) in WASSCE or NECO English Language, or an equivalent such as KCSE or IGCSE English (roughly CEFR level B1). You don't need the certificate: Phase 0 is free, so you can check for yourself
- A laptop running Linux, macOS, or Windows with WSL

How the course works

- A live cohort of up to 10 students, with a mentor.
- At least half your time is hands-on: every module ends with a lab.
- Live sessions are recorded, so you can rewatch them or catch up.
- Questions between sessions are answered within 24 hours on weekdays.
- A mentor reviews every phase project, like a pull request at work.

Phase 0 is free for everyone, before you enroll.

Curriculum

PHASE 0 · WEEKS 1–2 · FREE

Virtualization Fundamentals

Hypervisor types, CPU privilege levels, emulation versus virtualization, and setting up a lab with nested virtualization.

- Type 1 and type 2 hypervisors
- Privilege levels and trap-and-emulate
- Emulation vs virtualization
- Building a nested-virtualization lab

You build: Documented virtualization lab

PHASE 1 · WEEKS 3–4

QEMU Essentials

Machine types, CPU models, disk images, the QEMU monitor and QMP, and booting a Linux guest from the command line.

- Machine types and CPU models
- Disk images and qemu-img
- The monitor and QMP
- Booting a Linux guest

You build: Scripted VM launcher

PHASE 2 · WEEKS 5–6

Emulation with TCG

How QEMU translates guest code, user-mode emulation, and running ARM and RISC-V guests on an x86 machine.

- How TCG translates code
- User-mode emulation
- ARM and RISC-V guests

You build: Cross-architecture boot of an ARM Linux image

PHASE 3 · WEEKS 7–8

KVM Acceleration

Intel VT-x and AMD-V, the /dev/kvm API, vCPU threads, nested paging, and a minimal KVM program in C.

- VT-x and AMD-V
- The /dev/kvm API
- vCPU threads and the run loop
- Nested paging

You build: Tiny hypervisor that runs guest code through KVM

PHASE 4 · WEEKS 9–11

Virtio, Storage & Networking

virtio-blk and virtio-net, qcow2 images and backing chains, tap devices, bridges and vhost.

- virtio and paravirtualization
- qcow2 and backing chains
- tap devices and bridges
- vhost

You build: Multi-VM network with shared base images

PHASE 5 · WEEKS 12–13

libvirt & Automation

libvirt domains, virsh, storage pools, virtual networks and automation with the Python bindings.

- libvirt domains and virsh
- Storage pools and networks
- Automating with Python

You build: VM fleet provisioning tool

PHASE 6 · WEEKS 14–15

Live Migration & Snapshots

Pre-copy migration, dirty-page tracking, block jobs, snapshots and checkpoints.

- Pre-copy migration
- Dirty-page tracking
- Block jobs, snapshots and checkpoints

You build: Zero-downtime migration runbook

PHASE 7 · WEEKS 16–17

QEMU Internals & Device Development

The QEMU object model, the device model, MMIO and PCI, and writing a simple emulated device.

- The QEMU object model
- The device model
- MMIO and PCI
- Writing an emulated device

You build: Custom PCI device with a Linux driver

PHASE 8 · WEEK 18

Performance, VFIO & Security

CPU pinning, hugepages, NUMA, VFIO device passthrough, sVirt and guest isolation.

- CPU pinning, hugepages and NUMA
- VFIO passthrough
- sVirt and isolation

You build: Tuned VM with a benchmark report

PHASE 9 · WEEKS 19–20

Capstone

Build a virtualization platform feature in a team, to an enterprise brief.

- Design
- Build
- Demo and review

You build: Mini cloud that provisions, migrates and monitors VMs

Your certificate

You earn it by doing two things:

- Attend every live session. If you miss one, make it up by watching the recording and submitting that week's lab within 7 days. You can do this up to 2 times per course.
- Score more than 50% on the final evaluation at the end of the course.

Your Enterprise Readiness Rating is separate: it scores your projects, capstone and assessments so employers can see how job-ready you are.

Questions

Email academy@originclouds.io, or see the course online: https://academy.originclouds.io/courses/virtualization-engineer