

CLOUD

INTERMEDIATE

3 DAYS

KVM Virtualization: Administration, Tuning and Migration

A hands-on three-day programme covering the KVM, QEMU and libvirt stack end to end: host preparation, virtual machine lifecycle, storage and networking, CPU and memory tuning, snapshots and backup, live migration, and converting workloads from VMware and other hypervisors. Built against current distribution behaviour including modular libvirt daemons, external snapshots and the removal of SPICE.

Why KVM Virtualization?

The foundation under everything

Proxmox, OpenShift Virtualization, OpenStack and Nutanix AHV all sit on KVM. Learn the layer beneath.

Current, not recycled

Reflects modular libvirt daemons, external snapshots, VNC over SPICE and current machine types.

Built for the VMware exit

A full module on conversion, wave planning, rollback and what actually breaks in practice.

Command line first

virsh and domain XML taught properly, so you are not stranded when a GUI is unavailable.

What you'll be able to do

- ✓ Explain the KVM, QEMU and libvirt architecture and how the layers interact
- ✓ Prepare and validate a virtualization host including modular libvirt daemon configuration
- ✓ Create and manage virtual machines using virsh, virt-install and domain XML
- ✓ Configure storage pools and volumes across file, block, network and Ceph backends
- ✓ Choose disk formats, caching modes and provisioning strategy deliberately
- ✓ Build virtual networks using bridges, NAT, VLANs, Open vSwitch and passthrough
- ✓ Tune CPU topology, pinning, NUMA alignment and hugepages for predictable performance
- ✓ Implement external snapshots, incremental backup and tested recovery
- ✓ Perform live migration with and without shared storage
- ✓ Convert workloads from VMware and other hypervisors using virt-v2v
- ✓ Plan a migration in waves with rollback and realistic cutover windows
- ✓ Harden hosts and guests using sVirt, vTPM, Secure Boot and encryption
- ✓ Automate provisioning with Ansible, cloud-init and golden images
- ✓ Assess when to move from plain KVM to a managed platform

Who this is for

- Linux system administrators and infrastructure engineers
- VMware administrators moving to open source virtualization
- Platform, cloud and DevOps engineers
- Data centre and private cloud operations teams
- Architects planning a hypervisor migration

Curriculum

01 KVM Architecture and the Virtualization Stack

- KVM as a kernel module that turns Linux itself into the hypervisor
- Why every virtual machine is a Linux process, and what that gives you
- QEMU for device emulation and why it is driven through libvirt rather than directly
- libvirt as the management layer: drivers, domains and the API surface
- Hardware requirements, virtualization extensions and nested virtualization
- How the stack compares to a proprietary hypervisor suite, honestly
- **Hands-on:** verify host capability and inspect the running stack

02 Host Preparation and libvirt Configuration

- Package selection and enabling virtualization on the host
- Modular libvirt daemons against the deprecated monolithic daemon
- Socket activation, service dependencies and what changes when you switch
- Validating a host before it carries workloads
- System and session connections, and permission models for non-root use
- The web console as the supported graphical path
- **Hands-on:** build and validate a virtualization host from a clean install

03 Creating and Managing Virtual Machines

- Provisioning with virt-install and from existing images
- Domain XML anatomy: devices, buses, addresses and metadata
- Machine types and why current chipset selection matters for future support
- UEFI firmware, Secure Boot and firmware descriptor selection
- Installing Linux and Windows guests, including virtio driver injection
- Console and display access using VNC after the removal of SPICE
- Lifecycle operations, autostart and guest agent integration
- **Hands-on:** build Linux and Windows guests and edit their definitions by hand

04 Storage Pools, Volumes and Disk Formats

- Pool types: directory, LVM, iSCSI, NFS and Ceph RBD
- Choosing between qcow2 and raw, and current format versions
- Thin provisioning, allocation behaviour and space reclamation
- Cache modes, I/O modes and their effect on safety and throughput
- Discard and TRIM propagation to the underlying storage
- Resizing disks and filesystems safely, online and offline
- Block operations: commit, pull and copy for live disk management
- **Hands-on:** build multiple pool types and migrate a disk between them

05 Virtual Networking

- libvirt network modes: NAT, routed, isolated and bridged
- Building host bridges with NetworkManager now that virsh interface commands are gone
- Open vSwitch integration and when it earns its complexity
- VLAN tagging, trunking and multi-tenant separation
- virtio-net, multiqueue and driver selection after legacy NIC deprecation
- macvtap, SR-IOV and PCI passthrough for performance-sensitive workloads
- Firewalling, filtering and network isolation between guests
- **Hands-on:** build a bridged and a VLAN-segmented network end to end

06 CPU, Memory and Performance Tuning

- CPU models and the trade-off between passthrough performance and migration freedom
- Topology: sockets, cores and threads, and how guests interpret them
- vCPU pinning, emulator threads and isolating noisy neighbours
- NUMA topology alignment and why misalignment silently halves throughput
- Hugepages, transparent hugepages and memory backing choices
- Memory overcommit, ballooning and page sharing, with the risks stated
- I/O threads, queue depth and storage performance tuning
- **Hands-on:** benchmark a guest, tune it, and measure the difference

07 Snapshots, Backup and Recovery

- External snapshots as the supported mechanism, and why internal ones were retired
- Live snapshots, memory state and application consistency
- Guest agent quiescing for filesystem-consistent snapshots
- Merging, reverting and cleaning up snapshot chains
- Incremental backup using dirty bitmaps
- Backup tooling options and integrating with existing enterprise backup
- Testing restores rather than assuming them
- **Hands-on:** take, merge and restore snapshots, then run a full recovery

08 Live Migration and Availability

- Prerequisites: shared storage, network reachability and CPU compatibility
- Shared-storage against storage-copy migration
- Migration tuning: bandwidth limits, auto-converge and post-copy
- Diagnosing migrations that stall or fail to converge
- Clustering approaches, fencing and split-brain avoidance
- Designing for host maintenance without workload downtime
- **Hands-on:** live migrate running workloads under load, both ways

09 Migrating from VMware and Other Hypervisors

- The state of the VMware exit and what most estates are actually doing
- Converting guests with virt-v2v from VMware, Xen and Hyper-V sources
- Cold, warm and near-zero-downtime conversion approaches
- Windows guests: virtio driver injection and boot device changes
- Remapping networks, storage paths and resource reservations
- Wave planning, dependency mapping and cutover sequencing
- Rollback design, recoverability, and what commonly breaks: tools, licensing, MAC addresses, timing and monitoring
- **Hands-on:** convert and boot a foreign-hypervisor guest on KVM

10 Security, Hardening and Confidential Computing

- Mandatory access control and per-guest isolation on the host
- Least privilege for the virtualization service accounts
- Virtual TPM, Secure Boot and measured guest boot
- Disk encryption at rest and key management options
- Device passthrough risk and what direct hardware access exposes
- Host hardening, patch strategy and reducing reboot windows
- Confidential computing technologies and what they do and do not protect
- **Hands-on:** harden a host and a guest against a review checklist

11 Automation, Operations and Platform Choice

- Automating provisioning and configuration with Ansible against libvirt
- Guest customisation with cloud-init and template-driven builds
- Golden images, versioning and keeping templates current
- Monitoring, logging and capacity planning for a KVM estate
- Operational runbooks: patching, evacuation and incident handling
- When plain KVM is enough versus moving to a managed platform, compared against real requirements
- Total cost reality: the licence is free, the operational ownership is not
- **Hands-on:** automate a full host and guest build from code