Implementing and Interfacing with KVM

Cristian Palomo-Ramirez
cpalomo-ramirez@csumb.edu

Ben Ryan
bpryan@email.wm.edu

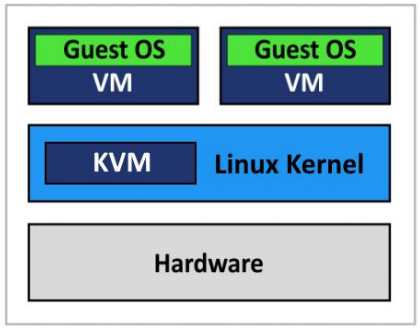
HPC Cluster Engineering Academy

LLNL-PRES-812987
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

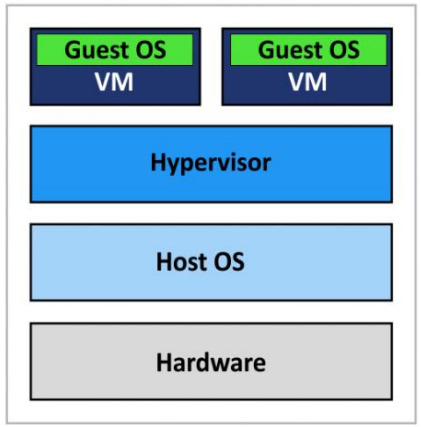
Lawrence Livermore National Laboratory

What is KVM and why is it useful?

- A Kernel-based Virtual Machine (KVM) is a type of VM that turns the Linux kernel into a bare-metal hypervisor.
- KVM allows the host machine to treat every guest (VM) as if it were a Linux process.
- Some of the benefits of a bare-metal VM are efficient usage of resources for smaller, specialized tasks, ease of testing, etc.
- The main benefits of KVM specifically is that it is built into Linux and is extremely efficient.



Type 1 Hypervisor
(Bare-Metal Architecture)



Type 2 Hypervisor
(Hosted Architecture)

Modified from Nakivo.com

Implementation and Tools

- Virt-manager is a GUI tool commonly used to manage KVM guests
- libvirt is the tool/package for managing guests via the command line
- Installing a variety of Linux distributions manually on individual guests is time consuming.
- Using tools such as BASH scripting and Ansible can automate this process.
  - Kickstart allows for pre-configuration of the operating system prior to installation.

High Level Goals

- Set up environment for testing MSR-safe kernel modules
- Create base images for various Linux distributions with configuration
  - Required packages
  - Test user with sudo privileges
  - Automated installation
  - Documentation for future admin use/maintenance
- Respond to Gitlab Continuous Integration requests
  - script/tool to run the CI request on the allocated image
  - capture results/logs
  - deallocate/clean up instance