Virtualization

Before Virtualization:

* Only one OS image per machine
* SW and HW tightly coupled
* Running multiple apps in same machine resulted in conflict
* Not all resources utilized
* Inflexible and costly

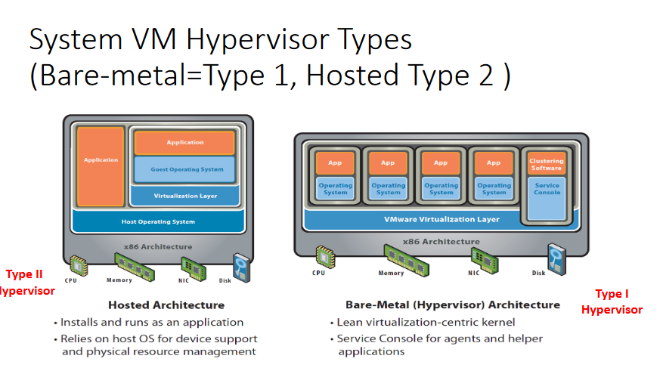
After Virtualization:

* HW independence of OS and Apps
* Virtual machines can be used to have more OS images
* Can manage OS and apps as single unit by encapsulating into these VMs

Defined:

* Virtualization is a technology that lets you create useful service using resources that are traditionally bound to HW. Allows to use physical machines full capacity by distributing its capabilities among many users/envs
* Can dedicate parts of hardware to different tasks, instead of having one hw for one task. This can reduce cooling and maintenance cost.
* Gist of it is that a machine has only one set of actual hw, so how can we get it to have more than one OS.
* Originally one layer HW one single OS, apps run on that OS.
  + When have devices in data center, one device can have multiple dif users.
    - Each user want dif config, os, virtualization PROVIDES THIS.

Architecture Interfaces:

* These interfaces at or near HW/SW are important for VM construction
  + ISA (Instruction Set Architecture)
    - Marks division between HW and SW, consists of Interface 4 and 3:
    - Interface 4 reps user ISA and includes aspects visible to an app program
    - Interface 3 reps system ISA, is a superset of user ISA and includes aspects visible only to OS SW responsible for managing HW resources
    - ISA defines supported data types, registers, hw support, i/o, etc. an abstract model of a CPU.
    - Specifies behaviour of machine code
  + Happens at intersection of 3 and 4. So OS translates whatever comes from top to hardware.
  + The boundary between your execution nHW and OS and HW and libraries.
  + ABI (Application Binary Interface)
    - Gives program access to HW resources and services available in system through user ISA and system call interface
    - Does not include system instructions, all app programs interact with hw resources indirectly by invoking os services via sys call interface.
    - EX: when use java VM helps run any Java prog on any os.
  + API (Application Programming Interface)
    - Gives program access to hw resources and services available in system through user ISA with exra support with high level language library calls
    - Have individual interfaces for dif apps.
    - So if providing extension for browser, if writing an extension the browser provides some things you can use to directly interact with libraries that support the app.
  + Process Virtual Machines:
    - Aka application VM
    - The virtualization below the API or ABI, providing virtual resources to single process executed on machine
    - Created for process alone, destroyed when process finished.
    - Something like JVM (on any OS we download the JRE and put it on top of the OS and so we can run any java program with that)
    - So when using Java we are using a VM
  + System Virtual Machines
    - Provide complete, persistent, system environment that supports OS along with many user processes. Provides guest OS with access to virtual HW resources like networking, I/O,a GUI with processor and memory.
    - Usually emulaltes an OS to provide platform to run programs where real hw not available to use
    - Virtualizing software translates the ISA used by one HW platform to another
    - SO Single host can run multiple isolated OS where the hardware managed by the VMM (virtual machine manager, or Hypervisor)
    - Have HW, virtualization SW, run Guest OS, then run App
      * 2 Types of Systen VM Hypervisor🡸 VM manager
        + Bare-Metal (Type 1)

This most of the cloud vendor envs.

Run directly on top of HW, and then can install multiple OS on top of that

**Basically no host OS between virtualization sw and HW**

The virtualization sw installed directly on HW where OS normally installed

* + - * + Hosted (Type 2)

On top of HW have host OS, and on top of host OS we install this virtual box manager, then install the guest os on top of that, then the app.

Runs with the OS of host machine. Additional OS can be installed

Most often used with end users and software testing

Big Players in Virtualization include:

* VirtualBox, VMware, Microsoft…