VIRTUALIZATION

OpenVZ

What is VIRTUALIZATION?

- Method of dividing computer resources
- Multiple isolated environments
- Types
  - Emulation
  - Paravirtualization
  - OS level virtualization

Operating System level virtualization

- Multiple isolated execution environments within a single OS kernel
- Doesn’t allow to run different kernels from different OSs at the same time
- Best possible performance
- Examples: OpenVZ, Linux-VServer, ..etc

Virtual Environment

- Look and feel of separate physical server
- VE has its own
  - Process tree with init
  - File system
  - Network interfaces with IP addresses, Firewall rules, Routing tables
  - ...etc

Virtual Environment

- Multiple VEs co-exist within a single physical server.
- Different VEs can run different Linux distros
- But, all VEs operate under same kernel

OpenVZ kernel

- Modified Linux based kernel which comes with following functionalities
  - Virtualization and Isolation
    - Enables many VEs within a single kernel
  - Resource management
    - Manages CPU, RAM, Disk space per VE basis
  - Checkpointing
    - Freezing a VE, Saving its state to disk with ability to unfreeze that state later

Virtualization and Isolation

- Each VE has its own set of resources provided by OS kernel
  - Files
    - System libraries, applications, virtualized /proc & /sys, virtualized lock ,.etc
  - users and groups
  - process tree
  - network
    - Allows VE to have its own IP addresses

Virtualization and Isolation

- Devices
  - a VE can be granted to access real device like network interfaces, serial ports
- IPC objects
  - shared memory
  - semaphores
  - messages

Resource Management

- Finite set of resources within a single kernel are shared among multiple VEs
- OpenVZ resource management subsystem consists of three components
  - Two-level disk quota
  - Fair CPU scheduler
  - UBC

Demo

Demo on OpenVZ

- Vzctl
  - create
  - start
  - set
  - restart
  - status
  - enter
  - stop
- Vzlist ..