

Virtual Machines and Vagrant

Benefits and Limitations

- Isolation - allows processes to run in contained environment
- Can be created and started quickly.
- Allows for repeatable experiences
- Better utilization of resources
- Typically will not execute as efficiently as host machine/host OS

Uses in Software Engineering

- Isolated and repeatable development environments
- Testing
- Deployment options
- Architectural prototyping
- Scalability
- Cloud Deployment

Virtual Machines

- Virtualize Hardware
- Emulation vs Virtualization
 - Emulation seeks to replicate how instructions are executed.
 - Virtualization seeks to replicate the functionality.
- Machine architecture or instruction set (x86/AMD vs ARM)

Hypervisors

- Type I - (Bare Metal) Run directly on the hardware with no host OS intervening. IBM first developed these in the 1960's
- Type II - Hypervisor is a program requiring a guest OS to access the host machine hardware.
- Note some tools like KVM can effectively convert a Linux box into a Type I

VM Creation

- Configure the hardware
 - CPU's (cores) - virtual cores are threaded in the host OS
 - Memory - is allocated from host OS, but host OS treats like normal program
 - Networks - provide various levels of isolation, e.g. host only, NAT
 - Disks - usually uses a file but can be granted access to dedicated physical devices
 - Displays - Windows in the host OS; can provide integration with host OS Windowing
 - Devices - USB, Cameras, input devices, etc can be exposed from host OS

Guest OS

- Most hypervisors support wide range of guest OS
- Can optimize elements of the virtualization for guest OS
- Hypervisors can run multiple OS's depending on host machine resources.
- Guest OS needs to be installed from media
- Guest OS's need to be maintained

Apple Silicon machines include Rosetta emulator for converting x86 instructions to Apple ARM instructions.

VM Orchestration

- Orchestration supports pre-configured VM's
- Reduces the management load for guest OS's
- Allows treating VM's like cattle rather than pets
- Resource sharing between host OS and guest OS's usually done with network protocols

Vagrant Orchestration

- Orchestration tools automatically allocate and configure infrastructure resources like VM's
- vagrant is an open source tool for orchestrating VMs
- Vagrant works with several VM providers - VMWare, Virtualbox, AWS EC2, and many others
- Vagrant directs the chosen VM provider to create a VM based on a specified VM image and can configure the VM once started
- Vagrant works with the VM provider to create virtual networks

Vagrant Box Files

- Vagrant relies on Box files to specify the VM system image
- Box files are provider-specific tarballs containing the VM disk image and a few configuration files used by Vagrant to configure the provider
- Vagrant maintains a central repository for Box file contributions but you can configure local repositories if preferred

Vagrant VM limitations

- Vagrant forces some conventions on the OS configurations in the Box images
 - Default users account vagrant with the password vagrant
 - The default vagrant account has su access
 - Public box images should have a default insecure ssh public key configured for the vagrant account

Vagrantfiles

- Vagrant looks for a Vagrantfile in the current directory and if not found walks the parent path looking for one
- The Vagrantfile specifies the box image to be used.
- The Vagrantfile is a ruby code fragment that configures the ruby config structure for the VM.
- The Vagrantfile can also specify specific scripts to be executed on the VM once it is booted. Vagrant refers to this as provisioning.

Booting a Vagrant VM

- The **vagrant up** command is used to start a VM.
- vagrant downloads the box file if needed, directs the hypervisor to create a VM with specified CPU, memory, etc.
- Vagrant directs the hypervisor to attach the disk image from the box file powers on the VM
- Once the VM is booted, vagrant logs in and replaces the insecure ssh key with a newly created key pair
- Once vagrant reports the VM is booted, login with **vagrant ssh**

If Box file specified by the Vagrantfile, is not locally available then vagrant automatically searches the default repository and downloads it. If the box file is present locally, it is still checked for updates in the repository

Note: While the default insecure vagrant ssh key file is replaced, vagrant does NOT reset the password for the default vagrant account.

Provisioning VM

- After Vagrant boots the VM and replaces the insecure ssh key, any provisioning steps can be run.
- Provisioning can run imbedded scripts, scripts within the VM, downloaded scripts, or configuration tools like Ansible or Chef.

Vagrant VM Networks

- Desktop hypervisors create a private virtual network for the VM and host
- The hypervisors provide a NAT gateway to allow the VM access to the host internet connection.
- Desktop hypervisors can enable the VM to participate directly on the host network (bridged network), but the VM should be hardened for the host network.

Hypervisor networks can configure private host-only networks by default. Only the host and the single VM can access the network. However, you can also configure shared private networks as well, which allow the multiple VM's to communicate on a private network.