BMC Software Inc.

Technical Disclosure Publication Document

**Authors**

Samir Bayani, Amit Modi and Navneet Waraich

**Hypervisor Agnostic Linux® Virtual Guest Customization Solution
Overview**

Posted: April 23, 2012

This document describes a methodology for a hypervisor-agnostic operating system customization solution for Linux® virtual guests.

# Background

With the exception of few hypervisor vendors, most hypervisors such as Citrix XenServer, RHEV, Hyper-V, etc. do not provide an out-of-the-box mechanism for customizing the identity (hostname, domain, etc) and network interface card settings (internet protocol (IP), domain naming service, etc.) of Linux Virtual Guests. An administrator is required to manually customize each virtual guest by logging in to the guest operating system using the hypervisor-provided console and executing pre-built scripts for customization. In a cloud-like environment where thousands of machines have to be commissioned every day, it is not feasible to manually log-in to each virtual machine and customize it. Some hypervisor platforms (such as VMWare) provide specific custom-built solutions and provide limited support. An alternate method that automates the complete customization process and provides complete control is hence highly desirable. This document describes an invention which can be used to automate the customization process for Linux Virtual Guests.

# Solution

Most hypervisors provide capability to attach media such as CD/DVD/Floppy/etc. to the virtual machine. It is also possible to configure the boot order to make virtual machine boot over such attached media. The proposed solution uses this capability to boot the virtual machine from a custom image, which consists of a Linux boot kernel and initrd (small file system which loads in ramdisk when machine is booted). The virtual machine when booted off this image triggers the init process. The init process in turn runs custom-built binaries to customize the guest-operating system. The below sections explain the complete invention.

## Custom Image Preparation:

- The guest customization image is comprised of a boot kernel and a custom-built initrd. This can be prepared from the same Linux distribution where initrd is prepared. The image could be an International Organization for Standardization (ISO) image, floppy image or a disk of any format supported by the hypervisor.
- A bootable image will be prepared on the Linux machine using a utility like mkisofs which generates an ISO. The image creation process will take the vmlinuz boot kernel and the initrd as inputs to output the image that will be used to boot the virtual machine.
- The custom initrd can be prepared using the Linux mkinitrd command on any Linux distribution such as CentOS. This will create a minimal version of a filesystem (an Initial

Ram Disk or initrd file) which can be used as a Linux boot environment loaded into ramdisk. An initrd file is usually (but not necessarily) a gzip compressed file.

- While doing this, all necessary drivers are injected into initrd image, required to detect disks of virtual machine, and to mount partitions present on the disks.
- The guest customization binary code will also be injected into initrd, and is responsible for performing actual customization.
- The initrd has an init script that is run when it boots. The init script will be modified to load correct drivers based on virtual machine hardware on boot, and run the guest customization commands.

## Customization process:

- The guest customization configuration file can be made available to the virtual machine via customization configuration file can be used to create a virtual media (such as CD/DVD/Floppy). This media can be attached to the virtual machine using hypervisor APIs.
- The guest customization image will be attached to the virtual machine as first boot device and Virtual Guest will be powered on.
- Once the virtual machine is booted off the guest customization image, it will load the boot kernel and initrd, which in turn will execute the guest customization binary.
- The binary will detect the disks on the virtual machine and search for /etc/fstab in ROOT partition to recognize the partition scheme of the machine.
- The binary will then mount all the disk partitions in a relocated root directory (e.g.; /tmp/simulated_root). The mounting will be done as per sequence defined in /etc/fstab file.
- On mounting relocated root, the complete file system of the virtual machine will be accessible in the relocated directory.
- The customization binary will read the customization configuration and update the operating system configuration files in relocated root directory as per the customization parameters.
- It will also configure the first boot script (rc.local) for all command-based customizations to be done when virtual machine is rebooted.
- Once the customization process is complete, the customization image will be detached from the virtual machine and the boot order will be reverted to its original state.
- The virtual machine will then boot over its primary hard disk and execute any first boot scripts if configured.
- The virtual machine is now available with the new configuration.

## Advantages of this approach:

- This will allow guest customization on hypervisor platforms which do not provide any mechanism for guest customization such as RHEV, Citrix Xen, etc.
- No agent or vendor-specific customization tools are required to be present in the source template from which virtual machine has been created.
- This approach has no dependencies on the network setting of the virtual machine operating system. The network based guest customization solutions are unusable in the following cases for the following reasons.
  - This is specifically useful in scenarios where a single source template is used to create multiple virtual machines as all of them may end up sharing the same hostname and IP address on boot. In such cases, the IP address, hostname conflict may occur, rendering the network-based solution unusable.
  - In case no network is configured in the source template operating system, the virtual machine will boot with no network connectivity.
- The approach is hypervisor agnostic. Only the part of attaching media and changing the boot order will be hypervisor specific. This capability is supported by all major hypervisors. This approach enables a management vendor such as BMC Software, Inc. to provide guest

Rev. 06/2009

customization support for newer hypervisor platforms quickly without having to deal with the vendor-specific solutions or build custom solutions for each hypervisor.

- This approach can also allow for standardizing the customization parameters across hypervisor platforms and allow customization of additional attributes that are not supported by existing vendor-specific solutions.
- This mechanism will also allow system administrators to execute additional first boot scripts if required.

**Drawings**
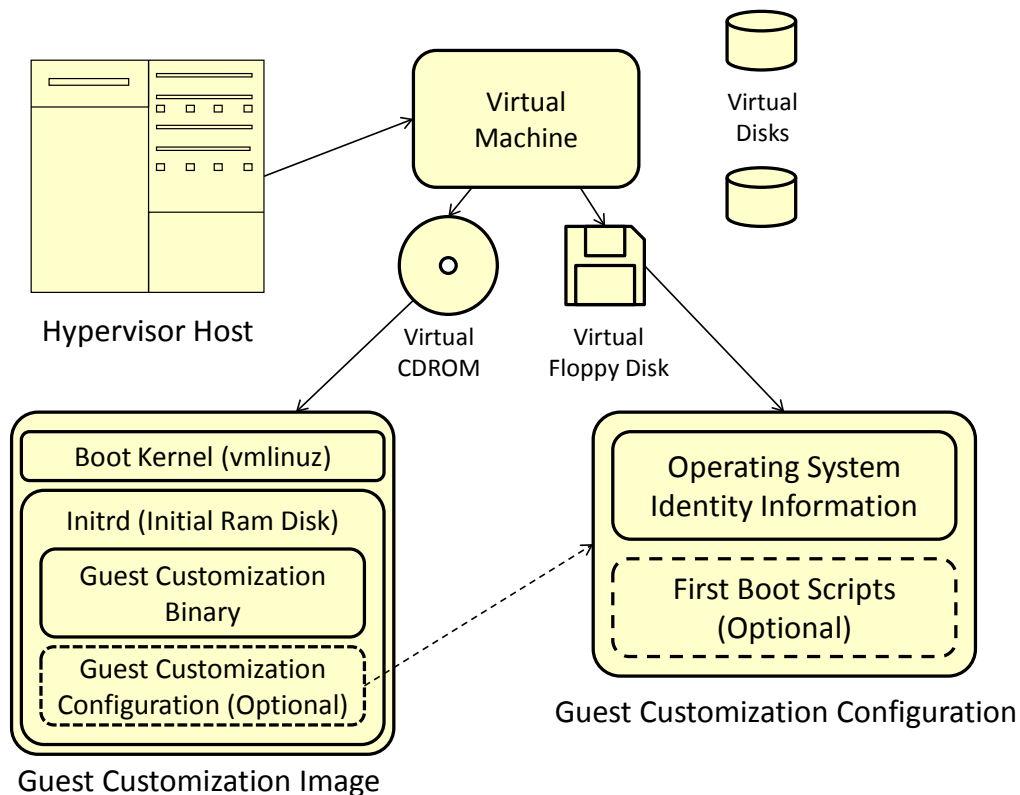


**Block diagram showing the various components involved in the interaction.**

Linux is the registered trademark of Linus Torvalds.

3