

vmm - Virtual Machine Manager

Overview

Virtual Machine Manager (vmm) is a bunch of wrapper scripts on top of existing Virtualization softwares. Vmm makes it easier to start and stop such machines and execute program (typically to perform tests) on these virtual machines.

Vmm is written in Bash and uses SSH to communicate/execute programs on the various machines.

Vmm is part of Free Software Client Reference System (FSCRS). FSCRS is a project at TIS Innovation Park in South Tyrol.

Contact info & Join us

Mailing list info@testingmachine.eu

Web site: <https://github.com/tis-innovation-park/vmm>

How to join: clone the repo, try it out – join the mailing list :)

Abbreviations

Machine - A virtual image as accessible by the supported virtualization softwares (e.g Virtualbox, qemu). A machine may be used by several clients (or none).

Client - A system configuration used by vmm to manage machines. Each client must belong to one (and only) one machine for the client. Separating clients from machines makes it easy for you to use one machine (the actual runnable virtual image) in many clients with different setups such as user, log files and timeouts.

Required software

- ssh (client)
- at least one virtualization software (see list of supported softwares below)
- bash

Supported Virtualization software

- Virtualbox
- qemu

We're looking into supporting: vmware,

Installation

- Download git code

```
git://github.com/tis-innovation-park/vvm.git
```

- Go to the vvm directory

```
cd vvm
```

- Configure the software

```
./configure --prefix <installationdir>
```

- Build the software

```
make
```

- Install the software

```
make install
```

- Verify the installation

```
<installationdir>/bin/vmm-client --list-clients
```

Setup

- Create the directory `$HOME/.vmm`

```
mkdir $HOME/.vmm
```

- Create `vmm.conf` in `.vmm`, typically with your favorite editor (emacs?)

```
emacs ~/.vmm/vmm.conf
```

In this file you can configure settings you want to use as default in your clients. It is perfectly possible to override these settings in your individual client configurations.

For a list of variables, see section Configuration syntax below.

Creating a machine

To create a machine `vmm` relies on the virtualization software. So if you want to manage a Virtualbox machine you (at least for now) create it with Virtualbox.

Creating a client

First of all you need to decide what machine you want to use with your client. In the example below we will assume it is called Debian6.0

- Create a directory for all clients:

```
mkdir ~/.vmm/clients
```

- Create a configuration file for the client (using a virtual machine):

```
[CLIENT NAME].conf
```

- Set the variables as you find suitable for your project.

- Example of a client configuration:

```
VM_NAME="Debian6.0"
VM_TYPE="VirtualBox"
VM_IP_ADDRESS=192.168.1.2
VM_USER=$USER
VM_SUPERUSER=root
SSH_PORT=22
SSH_SHUTDOWN_COMMAND="shutdown -h now"
```

For more variables see section Configuration syntax below

- Copy the public ssh key to the machine, e.g

```
ssh-copy-id 192.168.1.2
```

Using VVM

Client management

Command line options:

```
--list-clients - lists all configured clients
--start-clients client - starts client
--start-client-headless CLIENT_NAME - Start client called CLIENT_NAME
as headless (no screen)
--stop-clients client - stops client
--list-running-clients - Lists all clients currently running
--check-client-ssh - Checks if ssh is up on client
--check-client-status - Checks if clients is up and running
--client-exec CLIENT cmd - Exececute cmd on client
--client-exec-as-root - Exececute cmd on client as root
--client-x11 CLIENT - Checks if X11 is up and running on CLIENT
--client-screenshot CLIENT - Take a screenshot on CLIENT (not 100% ready)
--print-client CLIENT - Print the configuration for CLIENT
--wait-for-ssh CLIENT - Wait until ssh is up and running on CLIENT
--open-ssh CLIENT - Open an interactive shell (using ssh) on CLIENT
--check-client-online CLIENT - Check if CLIENT can ping the outside world
```

Machine management

```
--list-machines - lists all machines known to VVM
--start-machine client - starts machine
--start-machine-headless VM_NAME - Start machine called VM_NAME as
headless (no screen)
```

```
--stop-machine client - stops machine
--check-machine
--stop-client CLIENT_NAME - Stop client called CLIENT_NAME
--stop-machine VM_NAME - Stop machine called VM_NAME
```

Configuration syntax

The syntax for setting a variable is the same as in bash scripts (no coincidence!). Basically you write:

```
VARIABLE=VALUE
```

Variables

`LOG_FILE_DIR=/tmp/vmm/log` - sets the log file base directory to `/tmp/vmm/log`. This means that all logs can be found here.

`VM_STARTUP_TIMEOUT=10` - the time to wait for a virtual machine to start up before considering it to be 'dead'.

`VM_STOP_TIMEOUT=20` - the time to wait for a virtual machine to start up before taking more drastic actions to take down the machine. Ultimately VVM will take down a machine with a `kill`.

`SSH=ssh` - the SSH program to use

`SSH_TEST_OPTIONS=" -o connectTimeout=3"` - the options to pass to the SSH program (see SSH variable above) when testing of the machine is up (or not).

`VM_NAME="Debian6.0"` - the name of the machine to use in this client. It is important that your Virtualization software can find this machine.

`VM_TYPE="VirtualBox"` - what type of Virtualization software this machine belongs to. Currently you can use: VirtualBox and qemu

`VM_IP_ADDRESS=192.168.x.x` - the IP address of the virtual machine

`VM_USER=$USER` - the user you want to use when accessing the virtual machine

`VM_SUPERUSER=root` - the name of the super user/administrator account, typically used to reboot the machine

`SSH_PORT=22` - the port where the SSH server is running on the client

`SSH_SHUTDOWN_COMMAND="shutdown -h now"` - VVM will do its very best to shut down a machine as gracefully as possible. One way to do this is to try to shut it down using SSH. The command in this variable will be used to do that.

Example use:

```
/tmp/vmm-tmp-install/vmm-client -check-client-status $VBOXNAME  
/tmp/vmm-tmp-install/bin/vmm-client -check-client-status $VBOXNAME  
/tmp/vmm-tmp-install/bin/vmm-client -check-client-ssh $VBOXNAME  
/tmp/vmm-tmp-install/bin/vmm-client -client-exec $VBOXNAME "pkcs15-tool  
-L"
```