VMRC (Virtual Machine Repository and Catalog) Client

Instituto de Instrumentación para Imagen Molecular

Área de Grid y Computación de Altas Prestaciones

Universitat Politècnica de València

Camino de Vera S/N 46022 Valencia (SPAIN)

**Revisión** 08/04/2014

**Latest version of the document**: http://www.grycap.upv.es/vmrc

Content

[Pre-requisites 2](#_Toc319310602)

[Assumptions 2](#_Toc319310603)

[Deploying the VMRC Client 3](#_Toc319310604)

[Using the VMRC Client 4](#_Toc319310605)

[Adding a VMI to the catalogue without uploading the VMI files 4](#_Toc319310606)

[Uploading the VMI files to the entry of the catalogue 5](#_Toc319310607)

[List all the VMI entries in the catalogue 5](#_Toc319310608)

[Search the Most Appropriate VMIs 5](#_Toc319310609)

[Download the VMI image files to a local directory 6](#_Toc319310610)

[Removes an entry in the catalogue 6](#_Toc319310611)

[Managing Permissions to VMI entries 6](#_Toc319310612)

[Operations performed by the Administrator 7](#_Toc319310613)

[Add new Users to the VMRC catalog 7](#_Toc319310614)

# Pre-requisites

The binary version of the VMRC client has been compiled with Java JDK 1.7+. Therefore, it requires a JRE (Java Runtime Environment) 1.7+. The source version might be compiled with Java JDK 1.5+. Make sure you have the environment variable JAVA\_HOME correctly configured before attempting these steps.

# Assumptions

This document assumes that a VMRC server has been properly configured at the icaro.i3m.upv.es[[1]](#footnote-1) host. Therefore, the following URIs are available:

* VMRC Web GUI: http://icaro.i3m.upv.es:8080/vmrc-web-gui
* VMRC WS endpoint: http://icaro.i3m.upv.es:8080/vmrc/vmrc

Notice that if an HTTPS connector has been defined in Tomcat, then the following URLs can also be employed:

* VMRC GUI (HTTPS): https://icaro.i3m.upv.es:8444/vmrc-web-gui
* VMRC WS endpoint: https://icaro.i3m.upv.es:8444/ vmrc/vmrc

The vmrc-web-gui is not required for the client to operate. Only the server is required.

The command line client can seamlessly work with HTTP or HTTPS. Use whichever you want.

If the VMRC server and client are located on the same machine, you can safely use *localhost* in the previous URLs.

The VMRC client can seamlessly run from any platform with Java support. This includes GNU/Linux, OS X and Windows platforms. Being developed in Java, it should work on other platforms as well.

# Deploying the VMRC Client

1. Download the *vmrc-client-<version>.bin.tar.gz* from the aforementioned web site.
2. Unpack the package into a destination folder
   1. tar zxvf *vmrc-client-<version>.bin.tar.gz*

We will assume that the VMRC client has been uncompressed at the $HOME/vmrc-client-<version> folder.

1. Set the appropriate environment variables.
   1. Add at the end of the $HOME/.bashrc file the following lines:

|  |
| --- |
| export VMRC\_CLIENT\_LOCATION=$HOME/vmrc-client-<version>  export PATH=$PATH:$VMRC\_CLIENT\_LOCATION |

* 1. Reload this configuration
     1. source $HOME/.bash

1. Set the CLI scripts executable
   1. chmod u+x $VMRC\_CLIENT\_LOCATION/\*.sh
2. Setup your user credentials
   1. Create the file $HOME/.vmrc/vmrc\_auth with your credentials

|  |
| --- |
| johndoe:mypassword |

Notice that the johndoe user must exist in the VMRC server. If no valid credentials are supplied the VMRC client defaults to the “anonymous” user which has reduced privileges.

1. Verify that the remote VMRC Server catalog can be contacted.

Issue the following command:

|  |
| --- |
| ./vmrc.sh --list --uri $VMRC\_URL |

If no error arises, then the VMRC client has been successfully deployed.

In Windows, you can invoke the CLI as follows

|  |
| --- |
| java –jar vmrc-client.jar --list |

If you want to avoid specifying the VMRC\_URL in each CLI invocation, you can define the $HOME/.vmrc/vmrc.properties file with the following content:

|  |
| --- |
| vmrc.uri = https://icaro.i3m.upv.es:8444/vmrc/vmrc |

This way, the command can be issued as follows:

|  |
| --- |
| ./vmrc.sh --list |

# Using the VMRC Client

The following commands assume these environment variables properly defined with your specific configuration. In particular, remember that the machine icaro.i3m.upv.es does not exist:

|  |
| --- |
| export VMRC\_URL=https://icaro.i3m.upv.es:8444/vmrc/vmrc  export JAVA\_HOME=/usr/lib/jvm/java-6-openjdk/ |

## Adding a VMI to the catalogue without uploading the VMI files

This method creates a new entry in the catalogue to describe a VMI. This operation does not involve any file uploading to the VMRC repository.

|  |
| --- |
| ./vmrc.sh --add --file my\_vmi.vmi |

The file my\_vmi.vmi describes the VMI. This is a sample file:

|  |
| --- |
| system.name = MyImage7  system.hypervisor = vmware  system.location = /opt/vm\_images/dummy\_img.qcow2  cpu.arch = i686  disk.size = 5000  disk.os.name = Linux  disk.os.flavour = Ubuntu  disk.os.version = 11.15  disk.os.credentials.user = user2  disk.os.credentials.password = passwd2  disk.applications contains (name = com.mathworks.matlab, version = 8.0 )  disk.applications contains (name = net.nbcr.opal, version = 2.2 )  disk.applications contains (name = com.java, version = 1.6, path = /usr/local/bin/java ) |

## Uploading the VMI files to the entry of the catalogue

This method uploads the file related to a VMI into VMRC. VMware disks might be split in different files whereas KVM image files are just a single file. Therefore, VMWare disks should be compressed in a bundle (a single file) before uploading the file to VMRC.

|  |
| --- |
| ./vmrc.sh --upload $HOME/images/myvmi.img --vmi MyImage7 |

## List all the VMI entries in the catalogue

This method lists all the VMI entries in the catalogue that can be listed considering the credentials supplied by the client. It obtains an XML description of the VMI entries.

|  |
| --- |
| ./vmrc.sh --list |

## Search for the Most Appropriate VMIs

This method searches for the most appropriate VMIs in the VMRC catalogue that satisfy the requirements imposed by the user. Hard requirements will certainly be met by the VMI. Soft requirements will also be considered according to the user ranking. It obtains a ranked XML description of the (up to 10) VMIs that satisfy those requirements.

|  |
| --- |
| ./vmrc.sh --search req1.vmiq |

The specified file expresses the requirements that the VMI should met in order to be listed. Here comes a sample requirements file:

|  |
| --- |
| system.hypervisor = kvm  cpu.arch = i686  disk.os.name = Linux  disk.os.flavour = Ubuntu  disk.os.version >= 11.15  disk.applications contains (name = com.java, version >= 1.6)  soft 25 disk.applications contains (name = net.nbcr.opal, version > 2.0) |

By default, requirements are considered ‘hard ‘ and these must be satisfied by the VMI. The soft requirements can be ranked by the user. If you need further information about this language, please refer to specific document that describes it.

## Download the VMI image files to a local directory

This method downloads the specified VMI to a local directory in the client machine. The VMI should be stored in the VMRC repository. Otherwise, this command will fail.

|  |
| --- |
| ./vmrc.sh --download /tmp/my\_img.img --vmi MyImage7 |

## Removes a VMI from the catalogue

This method deletes an entry in the VMRC catalogue. If there is a related VMI image in the repository it also deletes it.

|  |
| --- |
| ./vmrc.sh --delete MyImage7 |

## Managing Permissions to VMI entries

Please refer to the VMRC Server document for further information about the permission model.

This is the syntax of the command:

|  |
| --- |
| ./vmrc.sh ---vmiAcl <vmi\_name> <operation> <perm> |

Where operation=[list | search | upload | download | delete | add ] and perm = [owner | all]

1. This is a fictional host. Do not try to use it. [↑](#footnote-ref-1)