

---

# RavadaVDI

Ravada Team

Apr 24, 2024



# ADMINISTRATOR DOCUMENTATION

<b>1</b>	<b>Ravada delivers</b>	<b>3</b>
<b>2</b>	<b>Who is Ravada meant for?</b>	<b>5</b>
<b>3</b>	<b>Ravada VDI documentation</b>	<b>7</b>
3.1	Install Ravada . . . . .	7
3.2	Install Ravada in Ubuntu . . . . .	8
3.3	Install Ravada in Debian . . . . .	10
3.4	Install Ravada on Fedora . . . . .	12
3.5	Install Ravada on Rocky Linux 9 or RHEL9 . . . . .	15
3.6	Install Ravada - Ubuntu Xenial . . . . .	18
3.7	Install Ravada from dockers . . . . .	19
3.8	Running Ravada in production . . . . .	21
3.9	log file . . . . .	23
3.10	Post Install Recomendations . . . . .	24
3.11	Development release . . . . .	25
3.12	Add KVM storage pool . . . . .	29
3.13	Apache . . . . .	30
3.14	How to import a Virtualbox image . . . . .	31
3.15	How to create a Virtual Machine . . . . .	33
3.16	How to dump a hard drive to Ravada . . . . .	33
3.17	How to Install a LDAP Server . . . . .	34
3.18	How to add a KVM template . . . . .	35
3.19	New ISO image . . . . .	36
3.20	How to import a OpenGnsys image . . . . .	37
3.21	Integrating Ravada and OpenGnsys . . . . .	39
3.22	Operation . . . . .	46
3.23	Swap Partition . . . . .	48
3.24	Troubleshooting frequent problems . . . . .	48
3.25	Upgrade Ravada . . . . .	53
3.26	Upgrade Ravada with Rollback . . . . .	54
3.27	Rollback Ravada Version . . . . .	56
3.28	SPICE client setup for Windows . . . . .	56
3.29	SPICE client setup for MacOS . . . . .	57
3.30	Virt-Viewer . . . . .	57
3.31	Associate SPICE files with remote viewer . . . . .	57
3.32	RemoteViewer . . . . .	58
3.33	Problems . . . . .	58
3.34	SPICE client setup for MacOS . . . . .	58
3.35	Virt-Viewer . . . . .	58

3.36	Associate SPICE files with remote viewer . . . . .	59
3.37	RemoteViewer . . . . .	59
3.38	Problems . . . . .	59
3.39	How to change the controller driver of a Windows VM to VirtIO . . . . .	60
3.40	Backup Virtual Machines . . . . .	60
3.41	Virtual Machine Manual Migration . . . . .	62
3.42	Kiosk Mode . . . . .	65
3.43	How to do a volatile Virtual Machine . . . . .	67
3.44	Ravada Cluster . . . . .	68
3.45	How to make a virtual machine disk sparse . . . . .	73
3.46	Server Hardware . . . . .	74
3.47	Cluster Hardware . . . . .	76
3.48	Ravada CLI . . . . .	76
3.49	Automatic Daily Operations . . . . .	78
3.50	Single Sign On . . . . .	79
3.51	AppAmor troubles . . . . .	80
3.52	Service Status . . . . .	81
3.53	Adding Custom Messages . . . . .	81
3.54	Create a custom login template . . . . .	81
3.55	Create a custom footer template . . . . .	83
3.56	Disable Spice Password . . . . .	83
3.57	Ravada advanced settings . . . . .	84
3.58	New documentation . . . . .	87
3.59	Configure LDAP Authentication . . . . .	89
3.60	Authentication with Active Directory . . . . .	91
3.61	Set access restrictions to Virtual Machines . . . . .	94
3.62	Test Active Directory . . . . .	96
3.63	Tuning Ravada . . . . .	97
3.64	Server Monitoring . . . . .	97
3.65	Guide button step-by-step . . . . .	100
3.66	Offline libraries in frontend . . . . .	102
3.67	Exposing a Virtual Machine . . . . .	102
3.68	Exposing Ports from a Virtual Machine . . . . .	103
3.69	Keeping the Base updated . . . . .	105
3.70	Spinoff Clone . . . . .	107
3.71	Live web Chatwoot . . . . .	110
3.72	PCI GPU Device Passthrough . . . . .	113
3.73	Network Bridge . . . . .	119
3.74	VirtioFS . . . . .	121
3.75	Firewall . . . . .	125
3.76	Best practices to install Linux Guests in Ravada . . . . .	126
3.77	Install Alpine Linux . . . . .	127
3.78	Install Windows 10 . . . . .	132
3.79	Install Windows 11 . . . . .	135
3.80	How to extend a Ravada Windows guest's disk space . . . . .	139
3.81	How to extend a Ravada Linux guest's disk space . . . . .	143
3.82	How to enable KVM virsh console access . . . . .	145
3.83	Reduce the image size after cloning a physical PC . . . . .	146
3.84	Qemu Guest Agent . . . . .	148
3.85	Set Hostname . . . . .	149
3.86	Development Tools . . . . .	150
3.87	Commit Rules . . . . .	151
3.88	Database Changes . . . . .	152
3.89	Editor configuration rules . . . . .	153

3.90	Local ISO server . . . . .	154
3.91	Run Ravada in development mode . . . . .	156
3.92	Testing environment . . . . .	156
3.93	How to create tests . . . . .	158
3.94	Localization and translation . . . . .	162
3.95	Ravada Documentation . . . . .	163
3.96	Frontend Libraries . . . . .	163
3.97	Hardening Spice security with TLS . . . . .	164



The chances are you're here because you've been searching for free Virtual Desktop Infrastructure (VDI) documentation. Whether it is a large or a small project, you can start with VDI and see its benefits right away! We assume you do want to start your VDI project as quickly as possible. Therefore, RAVADA VDI is the perfect software for you!

Ravada VDI is a free and open-source project that allows users to connect to a virtual desktop. So it is a VDI broker.





## RAVADA DELIVERS

By following the documentation [Read the Docs](#) and editing some configuration files, you'll be able to deploy a VM within minutes.



## WHO IS RAVADA MEANT FOR?

Ravada is meant for sysadmins who have some background in GNU/Linux, and want to deploy a VDI project.

---

**Note:** Get started with [VDI](#), without reinventing the wheel.

---

We have written some documentation and hosted it on [Read the Docs](#) for you. This documentation is *on-going*, so if there is something you think is missing, don't hesitate and drop us a line! In the meantime, we are still improving RAVADA VDI and its documentation, so new sections will be popping out from time to time.

Our code uses the [AGPL](#) license and it is [available on GitHub](#).



## RAVADA VDI DOCUMENTATION

The main documentation for the site is divided into three main sections:

- *Administrator Documentation*
- *Feature Documentation*
- *Guest VM section*

Do you feel like giving us a hand? Here you have all the information you need as *a developer*:

- *Developer Documentation*

### 3.1 Install Ravada

#### 3.1.1 Requirements

##### 3.1.2 OS

Ravada works in any Linux distribution but we only support the package for [Canonical Ltd. Ubuntu](#) , Debian and [Fedora](#) server.

You can also install Ravada using Docker.

##### 3.1.3 Hardware

It depends on the number and type of virtual machines. For common scenarios server memory, storage and network bandwidth are the most critical requirements.

##### Memory

RAM is the main issue. Multiply the number of concurrent workstations by the amount of memory each one requires and that is the total RAM the server must have. However, recent virtualization improvements allow you to overcommit the memory.

### Disks

The faster the disks, the better. Ravada uses incremental files for the disks images, so clones won't require many space. Read these [recommendations](#) if you want to buy a new dedicated server.

### 3.1.4 Install Ravada

Follow the detailed instructions in this section to install on different operating systems.

Ubuntu and Debian packages are built by the Ravada team:

- [Ubuntu](#)
- [Debian](#)

RPMS are kindly provided by *Eclipseo*.

- [RedHat Fedora](#)

Ravada can be installed from package or using docker. It is more suitable for development and testing:

- [Docker](#)

Follow [this guide](#) if you are only upgrading Ravada from a previous version already installed.

### 3.1.5 Client

The client must have a spice viewer such as virt-viewer. There is a package for linux and it can also be downloaded for windows.

### 3.1.6 Run

When Ravada is installed, learn [how to run and use it](#).

### 3.1.7 Help

Struggling with the installation procedure ? We tried to make it easy but let us know if you need [assistance](#).

There is also a [troubleshooting](#) page with common problems that admins may face.

## 3.2 Install Ravada in Ubuntu

### 3.2.1 Upgrade Ravada

Follow [this guide](#) if you are only upgrading Ravada from a previous version already installed.

### 3.2.2 Ubuntu

---

**Note:** We only provide support for Ubuntu 20.04 and 22.04.

---

We provide *deb* Ubuntu packages on the [UPC ETSETB repository](#).

Install the ravada package. Choose the one that matches your OS release:

- `ravada_2.2.1_ubuntu-18.04_all.deb`
- `ravada_2.2.1_ubuntu-20.04_all.deb` for Ubuntu 20.04 and Ubuntu 22.04

When you run `dpkg` now it may show some errors, it is ok, keep reading.

```
$ wget http://infoteleco.upc.edu/img/debian/ravada_2.2.1_ubuntu-20.04_all.deb
$ sudo apt-get update
$ sudo apt install ./ravada_2.2.1_ubuntu-20.04_all.deb
```

The last command will show a warning about missing dependencies. Install them running:

```
$ sudo apt-get update
$ sudo apt-get -f install
```

### 3.2.3 Mysql Database

#### MySQL server

**Warning:** MySql required minimum version 5.6

It is required a MySQL server, it can be installed in another host or in the same one as the ravada package.

```
$ sudo apt-get install mariadb-server
```

#### MySQL database and user

It is required a database for internal use. In this examples we call it *ravada*. We also need an user and a password to connect to the database. It is customary to call it *rvd\_user*. In this stage the system wants you to set a password for the sql connection.

Create the database:

```
$ sudo mysqladmin -u root -p create ravada
```

Grant all permissions on this database to the *rvd\_user*:

```
$ sudo mysql -u root -p ravada -e "create user 'rvd_user'@'localhost' identified by 'Pword12345*'"
$ sudo mysql -u root -p ravada -e "grant all on ravada.* to 'rvd_user'@'localhost'"
```

The password chosen must fulfill the following characteristics:

- At least 8 characters.
- At least 1 number.
- At least 1 special character.

### Config file

Create a config file at `/etc/ravada.conf` with the username and password you just declared at the previous step. Please note that you need to edit the user and password via an editor. Here, we present Vi as an example.

```
sudo vi /etc/ravada.conf
db:
  user: rvd_user
  password: Pword12345*
```

### 3.2.4 Ravada web user

Add a new user for the ravada web. Use `rvd_back` to create it. It will perform some initialization duties in the database the very first time this script is executed.

When asked if this user is admin answer *yes*.

```
$ sudo /usr/sbin/rvd_back --add-user admin
```

### 3.2.5 Client

The client must have a spice viewer such as `virt-viewer`. There is a package for linux and it can also be downloaded for windows.

### 3.2.6 Run

The Ravada server is now installed, learn [how to run and use it](#).

### 3.2.7 Help

Struggling with the installation procedure ? We tried to make it easy but let us know if you need [assistance](#).

There is also a [troubleshooting](#) page with common problems that admins may face.

## 3.3 Install Ravada in Debian

### 3.3.1 Upgrade Ravada

Follow [this guide](#) if you are only upgrading Ravada from a previous version already installed.



### 3.3.2 Debian

This is the guide to install Ravada in Debian 10 (Bullseye), Debian 11 (Buster) and Debian 12 (Bookworm).

We provide *deb* packages on the [UPC ETSETB repository](http://infoteleco.upc.edu/img/debian/ravada_2.2.1_debian-11_all.deb).

Install the ravada package. It is called *debian-11* but works fine in *debian-12* also.

- `ravada_2.2.1_debian-11_all.deb`

When you run `dpkg` now it may show some errors, it is ok, keep reading.

```
$ wget http://infoteleco.upc.edu/img/debian/ravada_2.2.1_debian-11_all.deb
$ sudo apt update
$ sudo apt install ./ravada_2.2.1_debian-11_all.deb
```

#### Debian KVM

You must enable spice KVM manually:

```
$ sudo ln -s /usr/bin/kvm /usr/bin/kvm-spice
```

### 3.3.3 Mysql/MariaDB Database

#### MariaDB server

It is required a MySQL or MariaDB server, it can be installed in another host or in the same one as the ravada package.

```
$ sudo apt-get install mariadb-server
```

#### MariaDB database and user

It is required a database for internal use. In this examples we call it *ravada*. We also need an user and a password to connect to the database. It is customary to call it *rvd\_user*. In this stage the system wants you to set a password for the sql connection.

```
$ sudo mysqladmin -u root -p create ravada
```

Grant all permissions on this database to the *rvd\_user*:

```
$ sudo mysql -u root -p ravada -e "create user 'rvd_user'@'localhost' identified by 'Pword12345*'"
$ sudo mysql -u root -p ravada -e "grant all on ravada.* to rvd_user@'localhost'"
```

The password chosen must fulfill the following characteristics:

- At least 8 characters.
- At least 1 number.
- At least 1 special character.

### Config file

Create a config file at `/etc/ravada.conf` with the username and password you just declared at the previous step. Please note that you need to edit the user and password via an editor. Here, we present Vi as an example.

```
sudo vi /etc/ravada.conf
db:
  user: rvd_user
  password: Pword12345*
```

### 3.3.4 Ravada web user

Add a new user for the ravada web. Use `rvd_back` to create it. It will perform some initialization duties in the database the very first time this script is executed.

When asked if this user is admin answer *yes*.

```
$ sudo /usr/sbin/rvd_back --add-user admin
```

### 3.3.5 Client

The client must have a spice viewer such as `virt-viewer`. There is a package for linux and it can also be downloaded for windows.

### 3.3.6 Run

The Ravada server is now installed, learn [how to run and use it](#).

### 3.3.7 Help

Struggling with the installation procedure ? We tried to make it easy but let us know if you need [assistance](#).

There is also a [troubleshooting](#) page with common problems that admins may face.

## 3.4 Install Ravada on Fedora

### 3.4.1 Requirements

### 3.4.2 OS

Ravada works in any Linux distribution.

---

**Note:** RPM packages are kindly built by a third party. Please check the release available. If you want the latest version it is advised to install it on top of Ubuntu or Debian.

---

### 3.4.3 Hardware

It depends on the number and type of virtual machines. For common scenarios are server memory, storage and network bandwidth the most critical requirements.

#### Memory

RAM is the main issue. Multiply the number of concurrent workstations by the amount of memory each one requires and that is the total RAM the server must have.

#### Disks

The faster the disks, the better. Ravada uses incremental files for the disks images, so clones won't require many space.

### 3.4.4 Install Ravada

Follow [this guide](#) if you are only upgrading Ravada from a previous version already installed.

### 3.4.5 Fedora and EPEL7

You can install ravada using the 'dnf' package manager.

```
$ sudo dnf install ravada
```

#### Add link to kvm-spice

This may change in the future but actually a link to kvm-spice is required. Create it this way:

```
$ ln -s /usr/bin/qemu-kvm /usr/bin/kvm-spice
```

#### MySQL server

It is required a MySQL server, in Fedora we use MariaDB server. It can be installed in another host or in the same as the ravada package.

```
$ sudo dnf install mariadb mariadb-server
```

And don't forget to enable and start the server process:

```
$ sudo systemctl enable --now mariadb.service
```

```
$ sudo systemctl start mariadb.service
```

### MySQL database and user

It is required a database for internal use. In this examples we call it *ravada*. We also need an user and a password to connect to the database. It is customary to call it *rvd\_user*. In this stage the system wants you to set a password for the sql connection.

**Warning:** If installing ravada on Ubuntu 18 or newer you should enter your user's password instead of mysql's root password.

Create the database:

```
$ sudo mysqladmin -u root -p create ravada
```

Grant all permissions on this database to the *rvd\_user*:

```
$ sudo mysql -u root -p ravada -e "grant all on ravada.* to rvd_user@'localhost' identified by 'Pword12345'"
```

The password chosen must fulfill the following characteristics:

- At least 8 characters.
- At least 1 number.
- At least 1 special character.

### Config file

Create a config file at */etc/ravada.conf* with the username and password you just declared at the previous step. Please note that you need to edit the user and password via an editor. Here, we present Vi as an example.

```
sudo vi /etc/ravada.conf
db:
  user: rvd_user
  password: Pword12345*
```

### 3.4.6 Ravada web user

Add a new user for the ravada web. Use *rvd\_back* to create it. It will perform some initialization duties in the database the very first time this script is executed.

When asked if this user is admin answer *yes*.

```
$ sudo /usr/sbin/rvd_back --add-user admin
```

### 3.4.7 Firewall (Optional)

The server must be able to send *DHCP* packets to its own virtual interface.

KVM should be using a virtual interface for the NAT domains. Look what is the address range and add it to your *iptables* configuration.

First we try to find out what is the new internal network:

```
sudo route -n
...
192.168.122.0    0.0.0.0          255.255.255.0  U        0          0          0 virbr0
```

So it is 192.168.122.0 , netmask 24. Add it to your iptables configuration:

```
$ sudo iptables -A INPUT -s 192.168.122.0/24 -p udp --dport 67:68 --sport 67:68 -j ACCEPT
```

To confirm that the configuration was updated, check it with:

```
$ sudo iptables -S
```

### 3.4.8 Client

The client must have a spice viewer such as virt-viewer. There is a package for linux and it can also be downloaded for windows.

### 3.4.9 Run

The Ravada server is now installed, learn [how to run and use it](#).

### 3.4.10 Help

Struggling with the installation procedure ? We tried to make it easy but let us know if you need [assistance](#).

There is also a [troubleshooting](#) page with common problems that admins may face.

## 3.5 Install Ravada on Rocky Linux 9 or RHEL9

### 3.5.1 Add Pre-Requisite Software

```
$ sudo dnf install mariadb-server
```

And don't forget to enable and start the server process:

```
$ sudo systemctl enable --now mariadb.service
```

```
$ sudo systemctl start mariadb.service
```

### MySQL database and user

It is required a database for internal use. In this examples we call it *ravada*. We also need an user and a password to connect to the database. It is customary to call it *rvd\_user*. In this stage the system wants you to set a password for the sql connection.

**Warning:** If installing ravada on Ubuntu 18 or newer you should enter your user's password instead of mysql's root password.

Create the database:

```
$ sudo mysqladmin -u root -p create ravada
```

Grant all permissions on this database to the *rvd\_user*:

```
$ sudo mysql -u root -p ravada -e "grant all on ravada.* to rvd_user@'localhost' identified by 'Pword1234567890!'"
```

### 3.5.2 Add Another Pre-Requisite Software

```
$ sudo dnf install httpd
```

Enable and Start HTTPD service

```
$ sudo systemctl start httpd
$ sudo systemctl enable httpd
$ sudo systemctl status httpd
```

Allow firewallD service

```
$ sudo firewall-cmd --permanent --add-port=80/tcp
$ sudo firewall-cmd --permanent --add-port=443/tcp
$ sudo firewall-cmd --reload
```

### 3.5.3 Requirements

```
$ sudo dnf install qemu-kvm libvirt virt-manager virt-install iptables-services httpd mysql-server perl
```

Install Required Perl Module using CPAN Shell Open CPAN shell using

```
$ perl -MCPAN -e shell
```

Install Required Perl Modules using inside CPAN shell using

```
$ install Authen::SASL Authen::ModAuthPubTkt Authen::Passphrase Authen::Passphrase::SaltedDigest Carp D
```

### 3.5.4 OS

Ravada works in any Linux distribution.

---

**Note:** RPM packages are kindly built by a third party. Please check the release available. If you want the latest version it is advised to install it on top of Ubuntu or Debian.

---

### 3.5.5 Hardware

It depends on the number and type of virtual machines. For common scenarios are server memory, storage and network bandwidth the most critical requirements.

#### Memory

RAM is the main issue. Multiply the number of concurrent workstations by the amount of memory each one requires and that is the total RAM the server must have.

## Disks

The faster the disks, the better. Ravada uses incremental files for the disks images, so clones won't require many space.

### 3.5.6 Make sure you are in root folder

```
$ cd /root
```

### 3.5.7 Download Ravada from Git Repo

```
$ git clone https://github.com/UPC/ravada.git
```

### 3.5.8 Install Ravada

Go to Ravada folder

```
$ cd ravada
```

Once Inside the Ravada folder, Install using make

```
$ make
```

```
$ make install
```

Once Ravada Perl module has been installed, confirm the file has been installed perl libraries by typing

```
$ ls /usr/local/share/perl5/5.32/
```

If you "Ravada" folder and all the lib folders installed, you have successfully installed Ravada Perl module

Now, it's time to copy essential files

```
$ cp -r /root/ravada /usr/share/ravada
```

```
$ cp -r /root/ravada/etc/systemd/* /etc/systemd/system/
```

```
$ cp /root/ravada/etc/ravada.conf /etc/
```

```
$ cp /root/ravada/etc/rvd_front.conf.example /etc/rvd_front.conf
```

```
$ sudo systemctl daemon-reload
```

Modify the rvd\_front.conf accordingly

Now, it's time to install rvd\_back service

```
$ perl /root/ravada/script/rvd_back
```

.

Once the rvd\_back is installed, we need to add the admin for the web interface: Add a new user for the ravada web. Use rvd\_back to create it. It will perform some initialization duties in the database the very first time this script is executed.

When asked if this user is admin answer *yes*.

```
$ sudo /usr/sbin/rvd_back --add-user admin
```

We can enable the rvd\_back and rvd\_front service

```
$ sudo systemctl daemon-reload
```

```
$ sudo systemctl enable rvd_back
```

```
$ sudo systemctl enable rvd_front
```

```
$ sudo systemctl start rvd_back
```

```
$ sudo systemctl start rvd_front
```

Change the Qemu config

```
$ vim /etc/libvirt/qemu.conf
```

Uncomment the following line:

```
$ save_image_format = "bzip2"
```

You have to restart libvirt after changing this file:

```
$ sudo systemctl restart libvirtd
```

### Add link to kvm-spice

This may change in the future but actually a link to kvm-spice is required. Create it this way:

```
$ ln -s /usr/bin/qemu-kvm /usr/bin/kvm-spice
```

Finally, we need to copy the xml template to the location below:

```
$ mkdir /var/lib/ravada
```

```
$ cp -r /root/ravada/etc/xml /var/lib/ravada/
```

Go ahead and restart rvd\_back, rvd\_front, and libvirtd to ensure everything is working as expected

```
$ sudo systemctl restart rvd_back
```

```
$ sudo systemctl restart rvd_front
```

```
$ sudo systemctl restart libvirtd
```

**Once everything goes as expected, you should be able to get to ravada web user-interface at:**

<http://your.ip:8081/> or <http://127.0.0.1:8081> if you run it in your own workstation.

## 3.6 Install Ravada - Ubuntu Xenial

It is advisable to install Ravada in one of the supported platforms: Ubuntu Bionic ( 18.04 ) or Fedora. But if you want to install in another distribution it can be done.

### 3.6.1 Packages

Install those packages:

```
$ sudo apt-get install perl libmojolicious-perl mysql-common libauthen-passphrase-perl libdbd-mysql-perl
```

In addition you need one package that it still may not be in Ubuntu repository, download from our own server at the [UPC ETSETB repository](#) and install it this way:

```
$ wget http://infoteleco.upc.edu/img/debian/libmojolicious-plugin-renderfile-perl_0.10-1_all.deb
```

```
$ sudo dpkg -i libmojolicious-plugin-renderfile-perl_0.10-1_all.deb
```

These packages are required to build some dependencies from source:

```
$ sudo apt-get install gcc gcc-4.8 make libssh2-1-dev libnet-ssh2-perl libssh2-1 libdate-calc-perl zlib
```



### 3.6.2 Perl Modules

Some Perl modules must be compiled from source:

```
$ sudo perl -MCPAN -we 'install "Net::SSH2"'
```

### 3.6.3 Database and configuration

From now on you can follow the instructions for Ubuntu 18.04. Skip to the MySQL installation step.

Install Ravada in Ubuntu 18.04 .

## 3.7 Install Ravada from dockers

### 3.7.1 Requirements

#### 3.7.2 OS

Install [Docker](#) and [docker-compose](#) on your local machine.

---

**Note:** There are several versions of the Compose file format – 1, 2, 2.x, and 3.x. For now, we use 2.2 keep this in mind <https://docs.docker.com/compose/compose-file/>

---

As of now[at the time of writing this doc], we recommend

```
$ docker --version
$ Docker version 10.7.0, build a872fc2f86
$ docker-compose --version
$ docker-compose version 0.8.0, build d4d1b42b
```

#### 3.7.3 Hardware

It depends on the number and type of virtual machines. For common scenarios are server memory, storage and network bandwidth the most critical requirements.

#### Memory

RAM is the main issue. Multiply the number of concurrent workstations by the amount of memory each one requires and that is the total RAM the server must have.

### Disks

The faster the disks, the better. Ravada uses incremental files for the disks images, so clones won't require many space.

### 3.7.4 Install Ravada from dockers

---

**Note:** Ravada source must be (locally) in: `~/src/ravada`

---

Follow this steps:

```
$ cd ~
$ mkdir src
$ cd src/
$ git clone https://github.com/UPC/ravada.git
$ cd ravada/dockerfy

$ docker-compose pull
$ docker-compose up -d
```

If it's not the first time and you want to start from scratch it's better to make sure to rebuild everything with:

```
$ docker-compose up -d --build --force-recreate
```

### 3.7.5 Ravada web user

Add a new user for the ravada web. Use `rvd_back` to create it. It will perform some initialization duties in the database the very first time this script is executed.

Connect to ravada-back docker: (We'll implement an automatically solution to avoid this case)

```
$ ~/src/ravada/dockerfy> docker exec -it ravada-back bash
$ root@6c3089f22e77:/ravada# PERL5LIB=./lib ./script/rvd_back --add-user admin
$ admin password: acme
$ is admin ? : [y/n] y
```

It's over! You can connect to: <http://localhost:3000>

### 3.7.6 Client

The client must have a spice viewer such as `virt-viewer`. There is a package for linux and it can also be downloaded for windows.

### 3.7.7 Run

The Ravada server is now installed, learn [how to run and use it](#).

### 3.7.8 Dockers troubleshoots

- Check if all dockers are up

```
$ docker-compose ps
```

- **No such file or directory**

If you see this message remember that the source project must be in your HOME directory inside src directory: ~/src/ravada

If you run *docker-compose* with *sudo* you may have problems with the code inside the docker. It is preferable to exec *docker-compose* with your user.

```
$ root@6f8d2946c40c:/ravada# PERL5LIB=./lib ./script/rvd_back --add-user soporte
```

```
$ bash: ./script/rvd_back: No such file or directory
```

- **Let's do a reset:**

We want to return to an initial starting point Remove all dockers and volume associated.

```
$ cd ~/src/ravada/dockerfy/utils
```

```
$ ./remove_all.sh
```

- **This type of credential error:** *docker.credentials.errors.InitializationError: docker-credential-secretservice not installed or not available in PATH [2171475] Failed to execute script docker-compose*

This package fix it, *golang-docker-credential-helpers*, you must install in your computer.

### 3.7.9 Help

Struggling with the installation procedure ? We tried to make it easy but let us know if you need [assistance](#).

Maybe this [slides](#) can help you.

There is also a [troubleshooting](#) page with common problems that admins may face.

## 3.8 Running Ravada in production

Ravada has two daemons that must run on the production server:

- *rvd\_back* : must run as root and manages the virtual machines
- *rvd\_front* : is the web frontend that sends requests to the backend

### 3.8.1 System services

#### Configuration for boot start

There are two services to start and stop the two ravada daemons:

After install or upgrade you may have to refresh the systemd service units:

```
$ sudo systemctl daemon-reload
```

Check the services are enabled to run at startup

```
$ sudo systemctl enable rvd_back
```

```
$ sudo systemctl enable rvd_front
```

### Start

```
$ sudo systemctl start rvd_back
$ sudo systemctl start rvd_front
```

### Status

You should check if the daemons started right the very first time with the status command. See troubleshooting frequently problems if it failed to start.

```
$ sudo systemctl status rvd_back
$ sudo systemctl status rvd_front
```

### Stop

```
$ sudo systemctl stop rvd_back
$ sudo systemctl stop rvd_front
```

## 3.8.2 Qemu

Ravada uses Qemu to manage the virtual machines. We encourage change this settings so hibernated machines are stored compressed. This way you will save a large amount of disk space on your server.

Edit the file `/etc/libvirt/qemu.conf` and uncomment and change this line:

```
save_image_format = "bzip2"
```

You have to restart libvirt after changing this file:

```
$ sudo systemctl restart libvirtd
```

## 3.8.3 Apache

You can reach the Ravada frontend heading to <http://your.server.ip:8081/>. It is advised to run an Apache server or similar before the frontend.

In order to make ravada use apache, you must follow the steps explained on [here](#).

## 3.8.4 Firewall

Ravada uses `iptables` to restrict the access to the virtual machines. These `iptables` rules grants access to the admin workstation to all the domains and disables the access to everyone else. When the users access through the web broker they are allowed to the port of their virtual machines.

In this example we restrict the access to the virtual machines display. The first line is optional but handy if we want to allow access to an administrator workstation for debugging purposes.

```
-A INPUT -p tcp -m tcp -s ip.of.admin.workstation --dport 5900:7000 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 5900:7000 -j DROP
```

Virtual machines display ports start on 5900. If you want to override this configuration, because some other firewall is filtering this port, you can do it this way:

Edit `/etc/libvirt/qemu.conf`, uncomment and change `remote_display_port_min`.

```
# Override the port for creating both VNC and SPICE sessions (min).
# This defaults to 5900 and increases for consecutive sessions
# or when ports are occupied, until it hits the maximum.
#
remote_display_port_min = 5910
#remote_display_port_max = 65535
```

Then restart libvirtd:

```
$ sudo systemctl restart libvirtd
```

The next virtual machine that starts will be shown from this port (5910). This won't change machines already started unless you shut them down and start them again.

### 3.8.5 Help

Struggling with the installation procedure ? We tried to make it easy but let us know if you need [assistance](#).

There is also a [troubleshooting](#) page with common problems that admins may face.

If you do not know how to create a virtual machine, please read [creating virtual machines](#).

## 3.9 log file

### 3.9.1 Frontend log file

You can set a custom log file for the frontend process in the configuration.

Edit the configuration file: `/etc/rvd_front.conf` and add or change next section:

```
,log => {
  log => 1
  ,file => '/var/log/ravada/rvd_front.log'
  ,level => 'debug'
}
```

Create the log directory and grant the Ravada user permissions to write to it:

```
root@server:~# mkdir /var/log/ravada
root@server:~# chown ravada /var/log/ravada
```

Now restart the Ravada front process and check the log file is created.

```
root@server:~# systemctl restart rvd_front
```

### 3.9.2 Security Logs

Security access will appear in the log file similar as an Apache log entry:

```
127.0.0.1 - [2020/09/01:12:20:32 +0200] GET /
```

Login failures show with an *access denied* message- Set the log level to *error* if you only want these:

```
Access denied to fff from 127.0.0.1
```

A fail2ban catch can be created from these. Contributions welcome.

## 3.10 Post Install Recomendations

### 3.10.1 Firewall

The server must be able to send *DHCP* packets to its own virtual interface.

KVM should be using a virtual interface for the NAT domains.

First we try to find out what is the new internal network:

```
sudo ip route
...
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1
```

So it is the interface virbr0.

Add it to your iptables configuration. This will allow some traffic between the host and the virtual machines: DHCP, DNS and ping.

```
$ sudo iptables -A INPUT -i virbr0 -p udp -m udp --dport 67:68 -j ACCEPT
$ sudo iptables -A INPUT -i virbr0 -p udp -m udp --dport 53 -j ACCEPT
$ sudo iptables -A INPUT -i virbr0 -p udp -m udp --dport 5353 -j ACCEPT
$ sudo iptables -A INPUT -i virbr0 -p tcp -m tcp --dport 53 -j ACCEPT
$ sudo iptables -A INPUT -i virbr0 -p tcp -m tcp --dport 5353 -j ACCEPT
$ sudo iptables -A INPUT -i virbr0 -p icmp -m icmp --icmp-type 8 -j ACCEPT
$ sudo iptables -A OUTPUT -o virbr0 -p udp -m udp --sport 67:68 -j ACCEPT
$ sudo iptables -A OUTPUT -i virbr0 -p udp -m udp --sport 53 -j ACCEPT
$ sudo iptables -A OUTPUT -i virbr0 -p udp -m udp --sport 5353 -j ACCEPT
$ sudo iptables -A OUTPUT -o virbr0 -p icmp -m icmp --icmp-type 8 -j ACCEPT
$ sudo iptables -A OUTPUT -o virbr0 -p tcp -m tcp -s 192.168.122.1/32 -j ACCEPT
```

To confirm that the configuration was updated, check it with:

```
$ sudo iptables -S
```

### 3.10.2 Configuration

The frontend has a secret passphrase that should be changed. Cookies and user session rely on this. You can have many passphrases that get rotated to improve security even more.

Change the file `/etc/rvd_front.conf` line *secrets* like this:

```
, secrets => ['my secret 1', 'my secret 2' ]
```

## 3.11 Development release

**Note:** If you are not sure, you probably want to install the stable release. Follow this [guide](#).

You can get the development release cloning the sources.

**Warning:** Don't do this if you install a packaged release.

```
$ git clone https://github.com/UPC/ravada.git
```

### 3.11.1 Possible development scenarios where to deploy

Obviously if you can deploy on a physical machine will be better but it is not always possible. In that case you can test on a nested KVM, that is, a KVM inside another KVM.

**Note:** KVM requires [VT-X / AMD-V](#).

```
$ sudo apt install cpu-checker
$ sudo kvm-ok
```

**Warning:** Do not consider [VirtualBox](#) in this situation, because it doesn't pass VT-X / AMD-V to the guest operating system.

### 3.11.2 Ubuntu required packages

Check this [file](#) at the line *depends* for a list of required packages. You must install it running:

**Note:** The `libvirt-bin` package was dropped since Ubuntu 18.10. The package was split into two parts: **libvirt-daemon-system** and **libvirt-clients**.

```
$ sudo apt-get install perl libmojolicious-perl mysql-common libauthen-passphrase-perl libdbd-mysql-perl
```

### 3.11.3 Mysql Database

Ravada needs a MySQL database to store information.

#### MySQL server

**Warning:** MySql required minimum version 5.6

It is required a MySQL server, it can be installed in another host or in the same one as the ravada package.

Ubuntu based distros

```
$ sudo apt-get install mysql-server
```

Debian based distros

```
$ sudo apt-get install mariadb-server
```

RedHat and Fedora based distros

```
$ sudo dnf install mariadb mariadb-server
$ sudo systemctl enable --now mariadb.service
$ sudo systemctl start mariadb.service
```

After completion of mysql installation, run command:

```
$ sudo mysql_secure_installation
```

#### MySQL database and user

It is required a database for internal use. In this examples we call it *ravada*. We also need an user and a password to connect to the database. It is customary to call it *rvd\_user*. In this stage the system wants you to set a password for the sql connection.

**Warning:** When installing MySQL you wont be asked for a password, you can set a password for the root user in MySQL via *mysql\_secure\_installation* or type your user's password when it ask's you for a password.

Create the database:

```
$ sudo mysqladmin -u root -p create ravada
```

Grant all permissions on this database to the *rvd\_user*:

```
$ sudo mysql -u root -p ravada -e "create user 'rvd_user'@'localhost' identified by 'Pword12345*'"
$ sudo mysql -u root -p ravada -e "grant all on ravada.* to rvd_user@'localhost'"
```

The password chosen must fulfill the following characteristics:

- At least 8 characters.
- At least 1 number.
- At least 1 special character.



### 3.11.4 Config file

Create a config file at `/etc/ravada.conf` with the username and password you just declared at the previous step.

```
db:
  user: rvd_user
  password: *****
```

When developping Ravada, your username must be able to read the configuration file. Protect the config file from others and make it yours.

```
$ sudo chmod o-rx /etc/ravada.conf
$ sudo chown your_username /etc/ravada.conf
```

### 3.11.5 Ravada web user

Add a new user for the ravada web. Use `rvd_back` to create it.

```
$ cd ravada
$ sudo PERL5LIB=./lib ./script/rvd_back --add-user admin
```

### 3.11.6 Firewall(Optional)

The server must be able to send DHCP packets to its own virtual interface.

KVM should be using a virtual interface for the NAT domains. Look what is the address range and add it to your iptables configuration.

First we try to find out what is the new internal network:

```
sudo route -n
...
192.168.122.0    0.0.0.0          255.255.255.0  U        0          0          0 virbr0
```

So it is 192.168.122.0 , netmask 24. Add it to your iptables configuration:

```
-A INPUT -s 192.168.122.0/24 -p udp --dport 67:68 --sport 67:68 -j ACCEPT
```

### 3.11.7 Client

The client must have a spice viewer such as `virt-viewer`. There is a package for linux and it can also be downloaded for windows.

### 3.11.8 Daemons

Ravada has two daemons that must run on the server:

- `rvd_back` : must run as root and manages the virtual machines
- `rvd_front` : is the web frontend that sends requests to the backend

Run each one of these commands in a separate terminal

Run the backend in a terminal:

```
$ sudo PERL5LIB=./lib ./script/rvd_back --debug
$ Starting rvd_back v1.2.0
```

The backend must be stopped and started again when you change a library file. Stop it pressing CTRL-C

Run the frontend in another terminal:

```
$ PERL5LIB=./lib morbo -m development -v ./script/rvd_front
$ Server available at http://127.0.0.1:3000
```

Now you must be able to reach ravada at the location <http://your.ip:3000/> or <http://127.0.0.1:3000> if you run it in your own workstation.

The frontend will restart itself when it detects a change in the libraries. There is no need to stop it and start it again.

### 3.11.9 Start/Shutdown scripts

If you wish to create a script to automatize the start and shutdown of the ravada server, you can use these two bash scripts:

```
1  #!/bin/bash
2  #script to start ravada server
3
4  display_usage()
5  {
6      echo "./start_ravada"
7      echo "./start_ravada 1 (messages not shown to terminal)"
8  }
9
10 if [ "$1" == "-h" ]
11 then
12     display_usage
13     exit 1
14 else
15     SHOW_MESSAGES=$1
16     export PERL5LIB="./lib"
17     if [ "$SHOW_MESSAGES" == "1" ]
18     then
19         morbo -m development -v ./script/rvd_front > /dev/null 2>&1 &
20         sudo PERL5LIB=./lib ./script/rvd_back --debug > /dev/null 2>&1 &
21     else
22         morbo -m development -v ./script/rvd_front &
23         sudo PERL5LIB=./lib ./script/rvd_back --debug &
24     fi
25     echo "Server initialized succesfully."
26 fi
27
```

```
1  #!/bin/bash
2  #script to shutdown the ravada server
3
4  pid_front=$(pidof './script/rvd_front')
5  if [ -n "$pid_front" ];then
6      echo "Shutting down rvd_front"
7      sudo kill -15 $pid_front
```

(continues on next page)

(continued from previous page)

```

8 else
9     echo rvd_front already down
10 fi
11
12 pid_back=$(pidof -x './script/rvd_back')
13 if [ -n "$pid_back" ];then
14     echo "Shutting down rvd_back"
15     sudo kill -15 $pid_back
16 else
17     echo rvd_back already down
18 fi

```

## 3.12 Add KVM storage pool

If you run out of disk space you may add a new disk.

---

**Note:** KVM must then be informed about this new space available by creating a new storage pool.

---

### 3.12.1 Add the drive to the system

After booting with the new drive, check `dmesg` to find out the name of the new disk. It will probably be called `/dev/sdSOMETHING`.

Double check this is actually the new disk, if not you may erase all the contents of the system. Type `df` to see the old disk partitions.

Create a new partition with `fdisk`. It should show it as empty. Add only one primary partition for all the free space.

Replace `sdX` by the real name of the new device:

```
$ sudo fdisk /dev/sdX
```

Format it with large files tuning:

```
$ sudo mkfs.ext4 -m 0.001 -T largefile /dev/sdX1
```

### 3.12.2 Mount the new partition

First find out the UUID. Copy the UUID value, use it for the next step.

```
$ blkid | grep sdX1
```

Add this new partition to the filesystem table:

```
$ sudo mkdir /var/lib/libvirt/images.2
$ sudo vim /etc/fstab
```

```
UUID=e5ff57d7-cde6-4117-ac5d-1b14ac5b218f /var/lib/libvirt/images.2 ext4 auto 0 3
```

It will mount it next time you boot, but it can be used without rebooting issuing:

```
$ sudo mount -a
```

### 3.12.3 Add the drive to the Virtual Manager

```
$ sudo virsh pool-define-as pool2 dir - - - - /var/lib/libvirt/images.2
$ sudo virsh pool-autostart pool2
$ sudo virsh pool-start pool2
$ sudo virsh pool-list
```

And that's it, now Ravada will use the pool that has more empty space the next time it needs to create a volume. If you want to fine tune what storage pool is used by default follow the [advanced settings documentation](#) .

## 3.13 Apache

You can set up any Web server on top of Ravada. In this doc we show how to run Apache as a proxy. Settings another tool like Nginx should be similar.

### 3.13.1 Upgrading

Even if you had Apache proxy already set up you must add some configuration options. Check Hypnotoad, modules and configuration and make sure it is exactly like this.

### 3.13.2 Configure Hypnotoad proxy

Hypnotoad is the engine that runs the web Ravada frontend. First of all you need to tell *hypnotoad* we are behind a proxy. This allows Mojolicious to automatically pick up the X-Forwarded-For and X-Forwarded-Proto headers.

Edit the file `/etc/rvd_front.conf` and make sure there is a line with *proxy => 1* inside hypnotoad.

```
hypnotoad => {
  pid_file => '/var/run/ravada/rvd_front.pid'
  ,listen => ['http://*:8081']
  ,proxy => 1
}
```

Restart the front server to reload this configuration:

```
$ sudo systemctl restart rvd_front
```

### 3.13.3 Install Apache

```
# apt-get install apache2
```

### 3.13.4 Enable apache modules

Enable these modules.

---

**Tip:** Do it even it is not the first time you set up Apache. We added some modules in the latest release.

---

```
# a2enmod ssl proxy proxy_http proxy_connect proxy_wstunnel headers
```

In this example we are using self signed certificate that comes with Apache SSL. You can request a real certificate from any vendor like *LetsEncrypt*.

### 3.13.5 Apache Proxy Configuration

Link the https configuration and add the proxy lines.

```
# a2ensite default-ssl
```

Edit `/etc/apache2/sites-enabled/default-ssl.conf`.

---

**Tip:** Do not forget new *ProxyPass* and *RequestHeader* lines added in the last release.

---

```
<IfModule mod_ssl.c>
  <VirtualHost _default_:443>
    ProxyRequests Off
    ProxyPreserveHost On
    ProxyPass /ws/ ws://localhost:8081/ws/ keepalive=On
    ProxyPass / http://localhost:8081/ keepalive=On
    ProxyPassReverse / http://localhost:8081/
    RequestHeader set X-Forwarded-Proto "https"
```

More information about SSL configuration from [Mozilla](#) and [Letsencrypt](#) non profit CA.

### 3.13.6 Apache redirect to https

Redirect all the connections to https.

Edit `/etc/apache2/sites-enabled/000-default.conf`

```
<VirtualHost *:80>
  ServerName hostname.domainname
  Redirect / https://hostname.domainname/
</virtualhost>
```

---

**Tip:** Remember restart Apache2 service, with `systemctl restart apache2` or `services apache2 restart`.

---

```
$ sudo systemctl restart apache2
```

## 3.14 How to import a Virtualbox image

---

**Note:** In this example we have VirtualBox machine called *EXAMPLE*.

---

### 3.14.1 Create an empty Virtual Machine

From the Ravada admin form, create a new virtual machine with the same operative system as the one installed in the virtual box machine.

Do not install anything in that machine, keep it off. Check what is the name of the disk volume and remove the other volumes.

Check the contents *file* attribute with the command `virsh edit EXAMPLE`,

```
source file='/var/lib/libvirt/images/EXAMPLE-vda-id8Q.img'/
```

Remove the swap, cdrom and other disk volumes.

This is the SWAP volume, notice its name ends in `.SWAP.img`.

```
<disk type='file' device='disk'>
  <driver name='qemu' type='raw' cache='none' />
  <source file='/var/lib/libvirt/images/EXAMPLE-aGam.SWAP.img' />
  <target dev='vdb' bus='virtio' />
  <address type='pci' domain='0x0000' bus='0x00' slot='0x08' function='0x0' />
</disk>
```

This is the cdrom disk drive, remove it too.

```
</disk>
```

Remove also the SWAP image file:

```
$ sudo rm /var/lib/libvirt/images-celerra1/EXAMPLE-_G_m.SWAP.img
```

### 3.14.2 Convert the image file

Make sure the VirtualBox machine is down, then convert the VDI to raw, then to qcow2.

This converted image wil be used by the empty virtual machine that was created before.

#### DIRECTLY VDI TO QCOW2

```
$ qemu-img convert -p -f vdi -O qcow2 EXAMPLE.vdi EXAMPLE.qcow2
```

#### OR IN TWO STEPS

##### 1. Convert to raw

```
$ VBoxManage clonehd --format RAW EXAMPLE.vdi EXAMPLE.img
```

## 2. Convert to qcow2

Convert to qcow2 using the name you saw before in the *XML* definition of the machine:

```
$ sudo qemu-img convert -p -f raw EXAMPLE.vdi -O qcow2 /var/lib/libvirt/images/EXAMPLE-vda-id8Q.img
```

## 3.15 How to create a Virtual Machine

It is probable that when you start reading this document, you have set up and install Ravada successfully and now it is time to create a virtual machine.

### 3.15.1 Steps

1. First of all, log into Ravada using the username and password you created in previous steps of installation.
2. on top right of the page, click on “Admin Tools” and then “Machines”.
3. In the new page, click on “New Machine”.
4. Choose a name for your virtual machine and choose an option for the ISO image. The selected image needs to be installed accordingly.
5. Now, you can see a list of available machines and the operations of each.

---

**Note:** Please note that you can see any upcoming error or problems in section “messages”.

---

## 3.16 How to dump a hard drive to Ravada

### 3.16.1 Introduction

### 3.16.2 Tools

- VMWare Converter. First you must register in [https://my.vmware.com/en/web/vmware/evalcenter?p=converter#tab\\_register](https://my.vmware.com/en/web/vmware/evalcenter?p=converter#tab_register)
- External hard disk or something similar, you will need to save about 60GB or 100GB.

### 3.16.3 Procedure

See the video: <https://www.youtube.com/watch?v=KtgdWsyNemA>

## 3.17 How to Install a LDAP Server

### 3.17.1 Install 389-ds

```
$ sudo apt-get install 389-ds-base
```

### 3.17.2 Configure directory server

#### Release 1.3 [old]

This is the configuration tool for older releases of 389 directory server. If there is no setup-ds tool in your system you probably have the new release, skip to Release 1.4 instruction below.

```
$ sudo setup-ds
```

When requested the server name, answer with the full qualified domain name of the host: `hostname.domainname`. In the next step you must supply the domain name as base for the configuration. So if your domain name is “`foobar.com`”, the base will be “`dc=foobar,dc=com`”.

#### Release 1.4 [new]

From release 1.4 we provide an example configuration file for creating the new directory instance.

```
[general]
config_version = 2
start = True
strict_host_checking = False

[slapd]
root_password = 12345678

[backend-userroot]
sample_entries = yes
suffix = dc=example,dc=com
```

After you set a password and correct suffix create a LDAP instance with *dscreate*:

```
$ sudo dscreate from-file ds389.conf
```

#### Enable and Start the service

```
$ sudo systemctl start dirsrv@localhost
$ sudo systemctl enable dirsrv@localhost
```



### 3.17.3 Add a LDAP section in the config file

The config file usually is `/etc/ravada.conf`. Add this configuration:

```
ldap:
  admin_group: test.admin.group
  admin_user:
    dn: cn=Directory Manager
    password: 12345678
    base: 'dc=example,dc=com'
```

Then restart the services:

```
$ sudo systemctl restart rvd_back
$ sudo systemctl restart rvd_front
```

### 3.17.4 Insert one test user

The ravada backend script allows creating users in the LDAP

```
$ sudo rvd_back --add-user-ldap jimmy.mculty
```

There are more commands to easily manage LDAP entries. Check the [LDAP section from the CLI documentation](#).

## 3.18 How to add a KVM template

ISO images are required to create KVM virtual machines. They can be placed or downloaded at run time.

### 3.18.1 Placing your own ISO image

Copy the .iso file to the KVM storage, it is `/var/lib/libvirt/images` by default. Make sure everybody can read it

```
# chmod 755 file.iso
```

Get the md5 for the ISO file, you will need it for the next step:

```
# md5sum file.iso
```

Add an entry to the SQL table:

```
mysql -u rvd_user -p ravada
mysql> INSERT INTO iso_images (name, description, arch, xml, xml_volume, md5, sha256, device)
VALUES ('name','the description', 'i386', 'name.xml' , 'name-vol.xml', 'bbblamd5sumjustgenerated'
```

### 3.18.2 XML file

A XML template file is required if you want to create machines from this ISO. In the directory `/var/lib/ravada/xml` there are examples. You can make new ones creating a new machine from another tool like `virt-manager`. Once it is down dump the xml with

```
# virsh dumpxml machine > name.xml
```

### 3.18.3 XML Volume file

Create a new xml volume file based in another one from `/var/lib/ravada/xml`.

### 3.18.4 URL based ISO (simplified)

For most Linux based distributions, you won't need to manually download the ISO. Here we're assuming that there are a valid VM definition XML and a volume XML files (based on Ubuntu 16.04 Xenial Xerus).

```
mysql -u rvd_user -p ravada
mysql> INSERT INTO iso_images (name, description, arch, xml, xml_volume, url, sha256_url)
      VALUES ('Mint 18.2 BETA Mate 64 bits', 'Mint Serena 18.2 BETA with Mate Desktop based on Ubuntu 1
```

### 3.18.5 Windows specifics

For Windows you will need the virtio ISO that can be downloaded from <https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/stable-virtio/virtio-win.iso>

Save it to `/var/lib/libvirt/images` and change the owner as you did for the Windows ISO.

```
# chmod 755 /var/lib/libvirt/images/virtio-win-0.1.126.iso
```

Then edit your Windows xml file and point the second CD drive to that ISO. For the current stable virtio version, it looks like this: `virsh edit machinename`

```
<disk type='file' device='cdrom'>
  <driver name='qemu' type='raw' />
  <source file='/var/lib/libvirt/images/virtio-win-0.1.126.iso' />
  <target dev='hdc' bus='ide' />
  <readonly />
  <address type='drive' controller='0' bus='1' target='0' unit='0' />
</disk>
```

You should also ensure that the system disk cache is set to 'directsync':

```
<driver name='qemu' type='qcow2' cache='directsync' io='native' />
```

If you're using the NEC xhci USB controller (the default one in our environment), you'll need to obtain a suitable driver for the `µPD720200` chipset. Plugable.com has it here <http://plugable.com/drivers/renesas> (2nd entry).

## 3.19 New ISO image

In order to use an ISO file when you create a new machine, you must first place it inside the KVM directory:

```
/var/lib/libvirt/images
```

Then you have to tell the storage engine that you changed a file manually.

```
sudo virsh pool-list
```

Name	State	Autostart
default	active	yes
pool2	active	yes

```
sudo virsh pool-refresh default
sudo virsh pool-refresh pool2
```

Reload the *new machine* form so the file you just uploaded shows up in the ISO list.

After that, Ravada is able to use the ISO when selecting it while creating a machine. Also, ISOs that were downloaded from Ravada can also be found in this directory.

If you want to include a KVM template instead, use this [guide](#).

If you need the xml or the volume (for when you wanna add a new template), you can extract them via virsh using the following commands:

```
$ sudo virsh pool-list #in order to see the list of available pools
$ sudo virsh vol-list default #in order to see the list of volumes available
$ sudo virsh vol-dumpxml --pool default name_of_the_existing_machine.qcow2 #obtain the dump of a selected volume
$ sudo virsh dumpxml name_of_the_existing_machine #obtain the xml of a existing selected machine
```

## 3.20 How to import a OpenGnsys image

First of all, copy the .img OpenGnsys image file to your Ravada system.

The .img OpenGnsys files are raw disk dump compressed with lzop. You can see the contents of a img lzop file with:

```
$ lzop -l B5part3dataUbuntu.img
$ method      compressed  uncompr.  ratio uncompressed_name
$ LZ01X-1      4536669058 7962881402 57.0% B5part3dataUbuntu.img.raw
```

Now, we will decompress the file. We have to force it because it doesn't have a .lzop extension:

```
$ lzop -x -S .img /opt/opengnsys/images/B5part3dataUbuntu.img
$ ls
$ 4 drwxr-xr-x 2 root      root          4096 Nov 27 12:49 .
$ 12 drwxrwxr-x 13 root      opengnsys      12288 Nov 27 12:48 ..
$ 7776256 -rwxrwxr-x 1 opengnsys opengnsys 7962881402 Oct 26 12:51 B5part3dataUbuntu
```

As you can see, the raw file has no extension.

```
$ file B5part3dataUbuntu
$ B5part3dataUbuntu: data
```

Now, we have the raw content of our image disk. Opengnsys uses partclone to create an image disk. The next step is dump this raw file to a qcow2 disk using partclone.

---

**Tip:** You can get the partclone utilities from opengnsys, you can download from the web: <https://partclone.org/download/>, or extract from a partclone package for your linux distribution.

---

You can inspect the raw file with:

```
$ ./partclone.info ./B5part3dataUbuntu
$ Partclone v0.2.38 http://partclone.org
$ unknow mode
$ File system:  EXTFS
$ Device size:   69.8 GB
$ Space in use:  69.7 GB
$ Free Space:    85.1 MB
$ Block size:    4096 Byte
$ Used block :   17008739
```

Now, we have to create an empty qcow2 file and dump the raw file inside. First of all, create the qcow2 file. It's important to check the size to ensure that the dump will fit in.

```
$ qemu-img create -f qcow2 B5part3dataUbuntu.qcow2 70G
```

Now, we mount the qcow2 file in your system, to dump it.

---

**Tip:** You can follow this guide to do it: [How to mount a qcow2 disk image](#)

---

```
$ qemu-nbd --connect=/dev/nbd0 ./B5part3dataUbuntu.qcow2
```

Now, we can create the partition structure of your disk. After create it, this is the result:

```
$ fdisk /dev/nbd0
$ Disk /dev/nbd0: 90 GiB, 96636764160 bytes, 188743680 sectors
$ Units: sectors of 1 * 512 = 512 bytes
$ Sector size (logical/physical): 512 bytes / 512 bytes
$ I/O size (minimum/optimal): 512 bytes / 512 bytes
$ Disklabel type: dos
$ Disk identifier: 0xc0545c3a
$
$ Device      Boot      Start         End      Sectors  Size Id Type
$ /dev/nbd0p1                2048 182454271 182452224   87G 83 Linux
$ /dev/nbd0p2      182454272 188743679   6289408    3G 82 Linux swap / Solaris
```

Now, we have 2 partitions, /dev/nbd0p1 and /dev/nbd0p2. To dump the img disk we have to use the partclone.ext3 utility:

Command to restore:

```
$ # ./partclone.ext3 -s ./B5part3dataUbuntu -O /dev/nbd0p1 -r
$ Partclone v0.2.38 http://partclone.org
$ Starting to restore image (./B5part3dataUbuntu) to device (/dev/nbd0p1)
$ Calculating bitmap... Please wait... done!
$ File system: EXTFS
$ Device size: 69.8 GB
$ Space in use: 69.7 GB
$ Free Space: 85.1 MB
$ Block size: 4096 Byte
$ Used block : 17008739
```

The process begin, and you can follow the logs:

```
$ 00:00:07, Remaining: 00:05:36, Completed: 2.04%, Rate: 12.16GB/min,
$
$ Elapsed 00:00:01, Completed: 99.97%, Rate: 1.23GB/min,
$ Elapsed: 00:56:28,
$ Remaining: 00:00:00, Completed: 99.98%, Rate: 1.23GB/min,
$ Elapsed: 00:56:29,
$ Remaining: 00:00:00, Completed:100.00%, Rate: 1.23GB/min,
$ Elapsed: 00:56:29, Remaining: 00:00:00,
$ Completed:100.00%, Rate: 1.23GB/min,
$
$ Total Time: 00:56:29, Ave. Rate: 1.2GB/min, 100.00% completed!
$
$ Total Time: 00:56:29, Ave. Rate: 1.2GB/min, 100.00% completed!
$ Syncing... OK!
```

```

$ Partclone successfully restored the image (./B5part3dataUbuntu) to the device (/dev/nbd0p1)
$ Cloned successfully.
$ root@willow: /ssd/estegoxCloneC6root@willow:/ssd/estegoxCloneC6#
$
$
$ Now, you can verify the filesystem, mounting it:
$ mount /dev/nbd0p1 /mnt/suse
$
$ ls -als /mnt/suse/
$ total 168
$  4 drwxr-xr-x  26 root root          4096 Mar  2 12:20 .
$  4 drwxr-xr-x   4 root root          4096 Mar  1 13:55 ..
$  4 drwxr-xr-x   2 root root          4096 Feb  3 2017 assig
$  4 -rw-----   1 root root           199 Mar  2 11:42 .bash_history
$  4 drwxr-xr-x   2 root root          4096 Feb  2 11:51 bin
$  4 drwxr-xr-x   4 root root          4096 Mar  2 12:30 boot
$  4 drwxr-xr-x   3 root root          4096 May 10 2017 mnt
$ 20 -rw-r--r--   1 root root       19732 Sep 23 2015 ogAdmLnxCli.log
$  4 drwxr-xr-x  80 root root          4096 Feb 19 11:33 opt
$
$ ...

```

Maybe didn't full the entire disk. You can expand it to fit all the disk:

```

$ umount /mnt/suse
$ e2fsck /dev/nbd0p1
$ e2fsck 1.43.5 (04-Aug-2017)
$ /dev/nbd0p1: clean, 1897474/5701632 files, 16969078/22806528 blocks
$ resize2fs /dev/nbd0p1

```

Now, unmount que qcow2 file:

```
$ qemu-nbd --disconnect /dev/nbd0
```

And that's all! Now you can create a Ravada vm and attach the disk.

It's possible that the system needs some extra adjustments. One typical problem is modify the `/etc/fstab` to change the `/dev/sda` references to `/dev/vda`. Another common problem is recreate the grub boot or add support to `/dev/vda` devices.

## 3.21 Integrating Ravada and OpenGnsys

Opengnsys is a open source project for remote deployment. This is a project developed for many Spanish universities to provide a full tool to deploy, manage, clone and manage remote computers. Opengnsys allow distribute and install many different operating systems.

Opengnsys is based in a PXE boot and a Linux graphical agent that allows manage remotely the computer from a centralized console. Here, we will explain how adapt our RAVA system to support boot from Opengnsys. The final objective is automate the creation a virtual machine with the same image that we have created for our classrooms.

### 3.21.1 DHCP boot options

First of all, we have to provide the dhcp options `next-server` and `filename.` to our dhcp server. Ravada is a KVM-based solutions, so, the dhcp server is the standard integrated in KVM. The DHCP-KVM server allows some configurations. Edit the KVM network configuration and add these options to the dhcp section:

```
$ virsh#virsh net-edit default
$ <network>
$ <name>default</name>
$ <uuid>85909d3b-f219-4055-92a3-d36c0c57810c</uuid>
$ <forward mode='nat' />
$ <bridge name='virbr0' stp='on' delay='0' />
$ <mac address='52:54:00:1a:06:50' />
$ <ip address='192.168.122.1' netmask='255.255.255.0'>
$ <tftp root='/ ' />
$ <dhcp>
$ <range start='192.168.122.30' end='192.168.122.254' />
$ <bootp file='grldr' server='<opengnsys-server-ip' />
$ </dhcp>
$ </ip>
$ </network>
```

`grldr` is the standard boot loader for Opengnsys.

### 3.21.2 Create and empty virtual machine

Now, you have to create and empty virtual machine. And empty machine boots from the iPXE network boot firmware client integrated in KVM. This is a snapshot of a vm booting process:

```
test-net-boot - Press SHIFT+F12 to exit (1) - Remote Viewer
File View Send key Help
SeaBIOS (version 1.10.2-1ubuntu1)
Machine UUID 1805fb4f-ca45-aaaa-bbbb-94124e739131

iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+3FF8FBA0+3FECFBA0 C980

Booting from ROM...
iPXE (PCI 00:03.0) starting execution...ok
iPXE initialising devices...ok

iPXE 1.0.0+git-20161027.b991c67+really20150424.a25a16d-1ubuntu2 -- Open Source N
etwork Boot Firmware -- http://ipxe.org
Features: DNS HTTP HTTPS iSCSI NFS TFTP AoE ELF MBOOT PXE bzImage Menu PXEXT

net0: 52:54:00:a7:49:34 using virtio-net on PCI00:03.0 (open)
[Link:up, TX:0 TXE:0 RX:0 RXE:0]
Configuring (net0 52:54:00:a7:49:34)..._
```

### 3.21.3 NAT adaptation

Now, we have detected that TFTP doesn't work with the default KVM NAT configuration. You have to add support for it. This document explain it: <https://beaveryoga.wordpress.com/2016/12/10/iptables-and-tftp-howto/>

### 3.21.4 Create the virtual machine in the Opengnsys console

We have to create the support configuration to this virtual PC in the Opengnsys console.

The virtual machine runs inside a NATed network, usually with a 192.168.122.0/24 IP address. Then, these vms uses the Ravada server as gateway. We have to create a new classroom with the NAT configuration to allow opengnsys to assign correctly the network mask and the gateway. This is the ravada-classroom configuration:

**Gestión Aulas**  
**Modificar**

Nombre	Aula-ravada
Ubicación	virtual
Proyector	<input type="checkbox"/>
Pizarra	<input type="checkbox"/>
Puestos	1
URL Imagen	aula.jpg <a href="#">Ver fotos</a>
Comentarios	
Dirección gateway del aula	192.168.122.1
Máscara de red del aula	255.255.255.0
IP Servidor DNS (opcional)	192.168.122.1
URL Servidor Proxy (opcional)	
Modo P2P -semillero durante la descarga-	peer
Tiempo P2P -semillero después de descarga-	600
Modo Multicast	Full-Duplex
IP Multicast	239.194.10.99
Puerto Multicast	9000
Velocidad Multicast	90
Propiedades aplicables a todos los ordenadores	
Menú	inlabVirtual
Repositorio	repoB5
Autoexec	
Perfil hardware	
Validación	No
Página login	
Página validación	

- gateway: 192.168.122.1 (KVM NAT default gateway)
- netmask: 255.255.255.0 (KVM NAT default netmask)
- IP multicast: your multicast group
- Menu: your page menu
- Repository: your image repository

Now, we have to create a computer inside your ravada classroom that is your virtual machine. Copy the MAC address of your empty machine:

```
$ virsh net-dhcp-leases default
```

\$ Expiry Time	MAC address	Protocol	IP address	Hostname	Client ID
\$ -----					
\$ 2018-11-27 09:11:39	52:54:00:a7:49:34	ipv4	192.168.122.178/24	-	01:52:54:

**Gestión Ordenadores**  
**Modificar**

Nombre \*

Dirección IP

Dirección MAC

Foto ordenador  [Ver fotos](#)

Perfil hardware

Repositorio

Menú

Autoexec

Interfaz de red

Driver de red

Validación

Página login

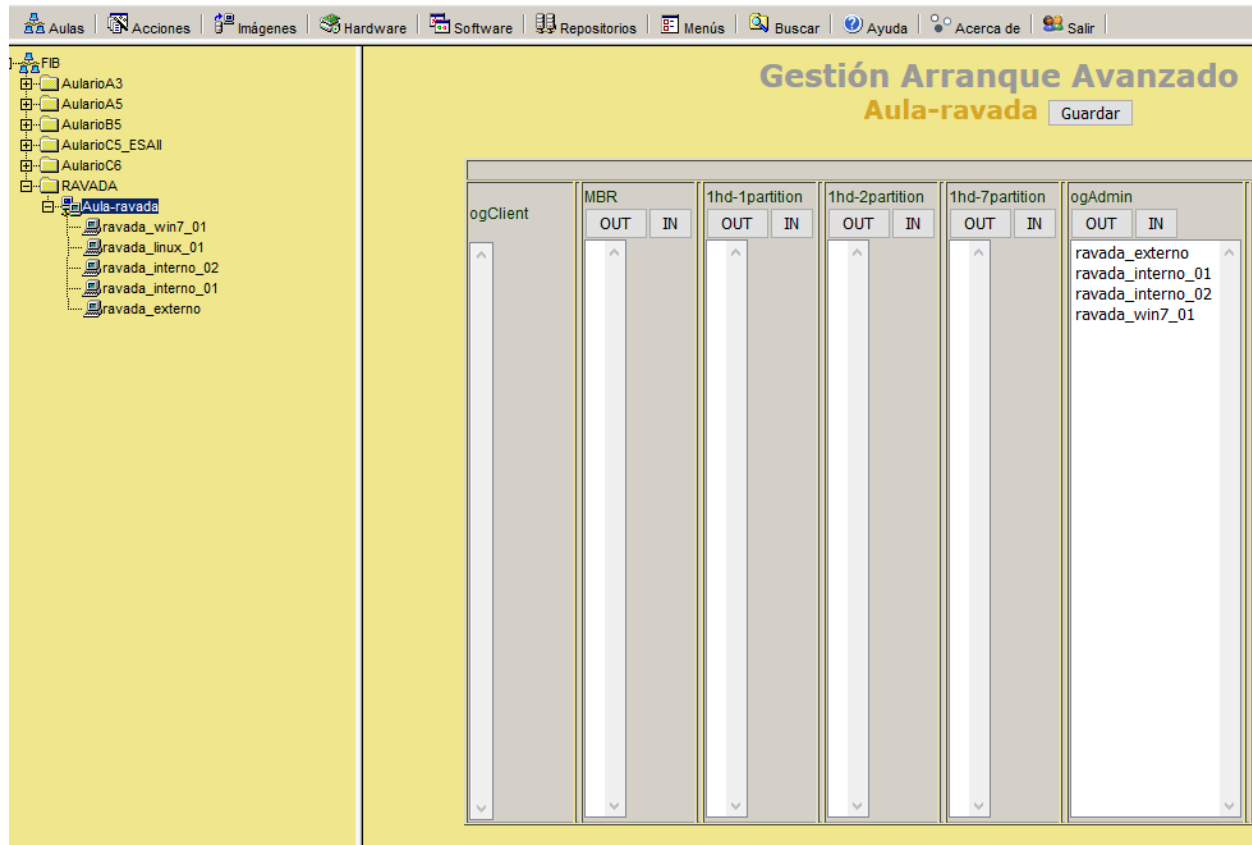
Página validación

\* AVISO: El nombre de equipo no debe superar los 15 caracteres si se instalan sistemas Windows.

Disco	Partición	Tipo	S.F.	S.O. Instalado	Tamaño (KB)	Imagen	Perfil Software	Contenido caché
Disco 1								
	1	LINUX	EMPTY		10485760			
	2	CACHE	CACHE		10484736			
	3	EMPTY	EMPTY		0			
	4	EMPTY	EMPTY		0			
	MSDOS				20971520			

And now, re-generate the TFTPBOOT files:





In this example, we have assigned the new PC to the ogAdmin group.

Now, you can boot the empty machine:

We have detected that the new machine boots, but it hangs just when the menu had to appear. After debugging, we have detected that the virtual machine don't have access to the http server with the menus. This a problem with routing. We have resolved creating a fake computer with the IP and MAC address of the KVM external NAT:

```
$ ifconfig
$ br0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
$ inet 10.10.73.24 netmask 255.255.255.0 broadcast 10.10.73.255
$ inet6 fe80::20a:f7ff:feba:c980 prefixlen 64 scopeid 0x20<link>
$ ether 00:0a:f7:ba:c9:80 txqueuelen 1000 (Ethernet)
$ RX packets 11251336 bytes 196755808380 (196.7 GB)
$ RX errors 0 dropped 0 overruns 0 frame 0
$ TX packets 11875794 bytes 4220061188 (4.2 GB)
$ TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

The screenshot shows the 'Gestión Ordenadores' (Manage Computers) interface with the 'Modificar' (Modify) tab selected. On the left is a tree view of the system structure, including folders like 'AularioA3', 'AularioA5', 'AularioB5', 'AularioC5\_ESAll', 'AularioC6', 'RAVADA', and 'Aula-ravada', with sub-items 'ravada\_externo' and 'ravada\_interno\_01'. The main area contains a form for editing the 'ravada\_externo' computer. The form fields are as follows:

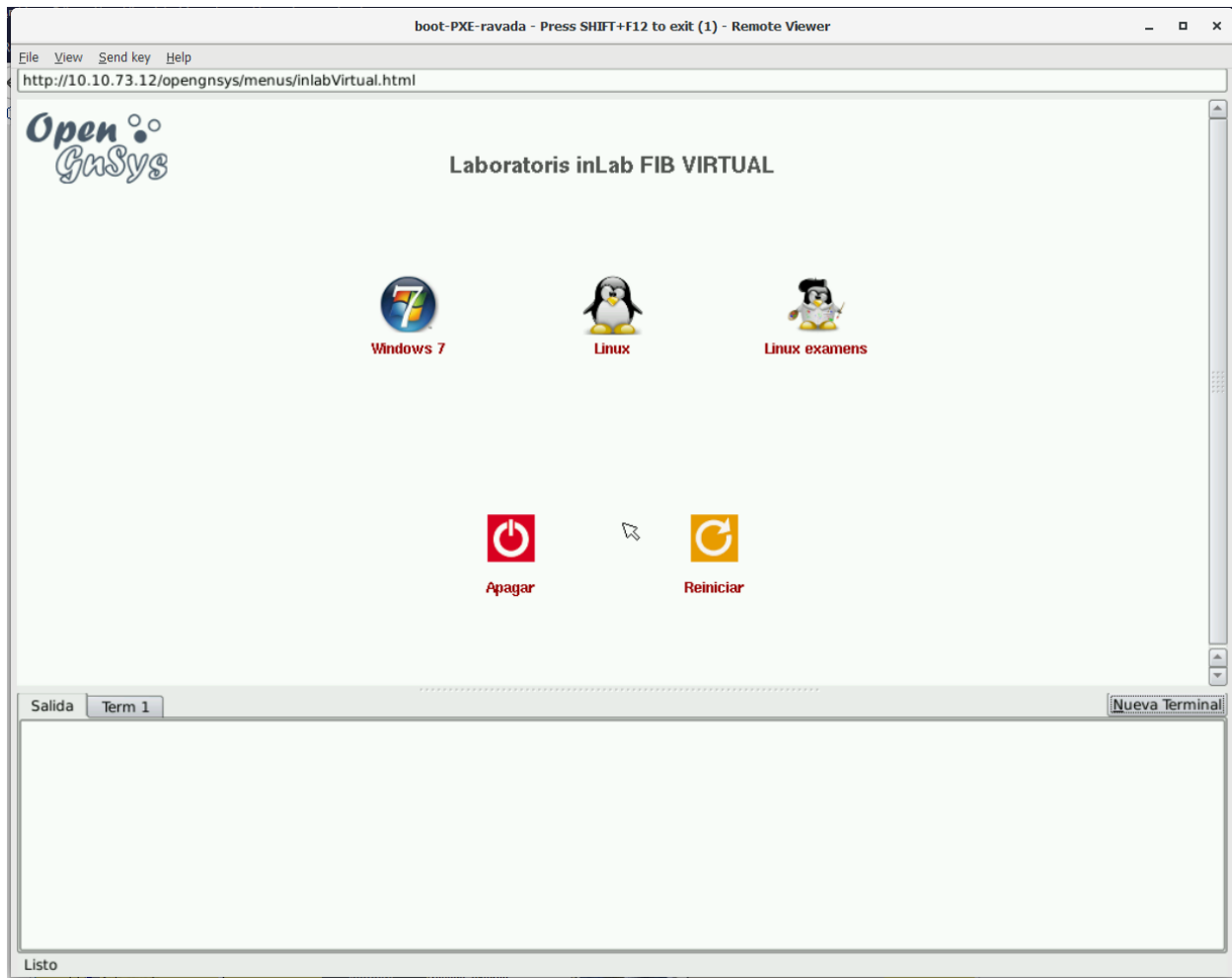
Nombre *	ravada_externo	 (150X110)-(jpg - gif - png) ---- Insertar foto> Examinar... No se ha seleccionado ningún archivo.
Dirección IP	10.10.73.24	
Dirección MAC	525400A74935	
Foto ordenador	fotoordenador.gif	<a href="#">Ver fotos</a>
Perfil hardware		
Repositorio	repoB5	
Menú	inLabB5	
Autoexec		
Interfaz de red	eth0	
Driver de red	generic	
Validación	No	
Página login		
Página validación		

\* AVISO: El nombre de equipo no debe superar los 15 caracteres si se instalan sistemas Windows.

At the bottom of the form are 'Cancel' and 'Accept' buttons. A status bar at the very bottom reads: 'Sin configuración: cliente no conectado al servidor.'

- IP: external NAT address of your RAVADA system
- MAC: external MAC address of your RAVADA system

This is our standard menu:



Now, you can boot your standard images in a virtual environment of Ravada. You have to be sure that your images have support to run in a virtualized system. In Linux images, the kernel have support `/dev/vda` devices. In Windows systems, you have to add the virtio drivers.

### 3.21.5 Special script adaptation

Our images boots ok, but our opengnsys instance doesn't detect the virtual disk. The problem was in our system, wich is very old (v1.0.5). To add support to detect `/dev/vda` devices, we have patched the `/opt/opengnsys/client/lib/engine/bin/Disk.lib` library:

```
$ # Listar dispositivo para los discos duros (tipos: 3=hd, 8=sd 253=vda). inLab 2018
$ ALLDISKS=$(awk '($1==3 || $1==8 || $1==253) && $4!~/[0-9]/ { printf "/dev/%s ",$4' /proc/partitions)
$ VOLGROUPS=$(vgs -a --noheadings 2>/dev/null | awk 'printf "/dev/%s ",$1')
$ ALLDISKS="$ALLDISKS $VOLGROUPS"
```

This patch adds vda disk detection to the `ogDiskToDev` function. (minor 253 -> vda devices). This problem was fixed in later versions.

## 3.22 Operation

---

**Note:** Must run from the command line interface (CLI)

---

### 3.22.1 Create users

```
$ sudo ./usr/sbin/rvd_back --add-user=username  
$ sudo ./usr/sbin/rvd_back --add-user-ldap=username
```

### 3.22.2 Import KVM virtual machines

Usually, virtual machines are created within ravada, but they can be imported from existing KVM domains. Once the domain is created :

```
$ sudo ./usr/sbin/rvd_back --import-domain=a
```

It will ask the name of the user the domain will be owned by. You must enter a valid username that will own the virtual machine within Ravada.

Also you will be asked whether you want to spinoff the disk volumes or not.

You probably want to answer **no**. You should answer *yes* if you created the volumes with a backend file and want to flatten them out just. It may be the case if you migrated the virtual machine from another server.

### 3.22.3 View all rvd\_back options

In order to manage your backend easily, rvd\_back has a few flags that let you make different things (like changing the password for an user).

If you want to view the full list, execute:

```
$ sudo rvd_back --help
```

### 3.22.4 Admin

---

**Note:** Must run from the frontend

---

#### Create Virtual Machine

Go to Admin -> Machines and press *New Machine* button.

If anything goes wrong check Admin -> Messages for information from the Ravada backend.

## ISO MD5 mismatch

When downloading the ISO, it may fail or get old. Check the error message for the name of the ISO file and the ID.

### Option 1: clean ISO and MD5

- Remove the ISO file shown at the error message
- Clean the MD5 entry in the database:

```
$ mysql -u rvd_user -p ravada mysql > update iso_images set md5='' WHERE id=*ID*
```

Then you have to create the machine again from scratch and make it download the ISO file.

### Option 2: refresh the ISO table

If you followed *Option 1* and it still fails you may have an old version of the information in the *isoimages* table. Remove that entry and insert the data again:

```
$ mysql -u rvd_user -p ravada -e "DELETE FROM iso_images WHERE id=_ID_"
```

Insert the data from the SQL file installed with the package:

```
$ mysql -u rvd_user -p ravada -f < /usr/share/doc/ravada/sql/data/insert_iso_images.sql
```

It will report duplicated entry errors, but the removed row should be inserted again.

## 3.22.5 Create base of a Virtual Machine

Go to Admin tools -> Virtual Machines

### 1st Base

If you have configured your Virtual Machine, now you can do the Base:

- Select the Base checkbox.

The Virtual Machine will be published if you select the Public checkbox.

### 2nd base or more

In this case, you have a previous Base and you've made some changes at the machine. Now you must prepare a Base again.

Steps:

1. Remove all clones of this Virtual Machine.
2. Select the Base checkbox to prepare base.

## 3.23 Swap Partition

Though installing the Operating System in a Virtual Machine may be the same as in real machines, carefully planning the swap partition of the virtual machines will save a lot of disk space. Follow those guidelines.

### 3.23.1 Swap Volume

Mark at the creation of the Virtual Machine that swap disk volume. The desired max size must be declared there. That way a different disk will be created with that purpose. This volume is different than regular data disk volumes: it will be created only at the start of the machine and it will be destroyed at shutdown. Also, this volume won't keep incremental changes from the base, as data volumes do.

### 3.23.2 Partitioning

Later on we will address particular considerations for swap space in different operating systems. By now, keep in mind that the best practice is to keep a disk volume *only for swap*.

### 3.23.3 Linux

It is recommended keep the swapping the less possible. If possible remove the swap partitions and the swap configuration in `/etc/fstab`.

Some software on Linux requires some swap to run. If so, set the *swappiness* to the minimum this way:

```
$ sudo sysctl vm.swappiness=1
```

To make this change permanent add it to the file: `/etc/sysctl.conf`

## 3.24 Troubleshooting frequent problems

### 3.24.1 Could not access KVM kernel module:

The system shows this message on trying to start a virtual Machine:

```
Could not access KVM kernel module: Permission denied failed to initialize KVM:␣  
↪Permission denied
```

That means the host has no virtual capabilities or are disabled. Try running:

```
$ sudo tail -f /var/log/syslog  
$ sudo modprobe kvm-intel
```

If it shows a message like this it means the BIOS Virt feature must be enabled:

```
kvm: disabled by bios
```

or try: `kvm-ok` command

```
kvm-ok  
INFO: /dev/kvm does not exist  
HINT:  sudo modprobe kvm_intel  
INFO: Your CPU supports KVM extensions
```

INFO: KVM (vmx) is disabled by your BIOS  
 HINT: Enter your BIOS setup and enable Virtualization Technology (VT),  
 and then hard poweroff/poweron your system  
 KVM acceleration can NOT be used

### 3.24.2 Dealing with permissions

The system may deny access to some directories.

#### On Screenshots ( requires review )

That problem showed up in Vanilla Linux 4.10.

When running the screenshot command it returns:

```
failed to open file '/var/cache/libvirt/qemu/qemu.screendump.31DvW9': Permission denied
```

#### Apparmor

At the file : /etc/apparmor.d/usr.lib.libvirt.virt-aa-helper

```
/var/cache/libvirt/qemu/ rw,  
/var/cache/libvirt/qemu/** rw,
```

### 3.24.3 Error with MySQL version < 5.6

For example the following message:

```
DBD::mysql::db do failed: Invalid default value for 'date_send' at /usr/share/perl5/  

↳ Ravada.pm line 276.
```

DEFAULT CURRENT\_TIMESTAMP support for a DATETIME (datatype) was added in MySQL 5.6.

Upgrade your MySQL server or change: datetime for timestamp

```
date_send datetime default now(),
```

More information [about](#).

### 3.24.4 MySQL binlogs fill the disk

If you are using MySQL it may create a lot of binlog files that in some cases may fill your disk drive. If you are not using replication it is safe to disable them if it is only used for Ravada.

Edit the file /etc/mysql/mysql.conf.d/mysqld.cnf comment the binlog lines and add *skip-log-bin=true* :

```
[mysqld]  
  
skip-log-bin = true  
# binlog_expire_logs_seconds      = 2592000  
# max_binlog_size      = 100M
```

To purge the logs run the mysql client and type:

```
mysql> purge binary logs before '2024-11-15';
```

### 3.24.5 Spice-Warning Error in certificate chain verification

```
(/usr/bin/remote-viewer:2657): Spice-Warning **: ssl_verify.c:429:openssl_verify: Error_
→ in certificate chain verification: self signed certificate in certificate chain_
→ (num=19:depth1:/C=IL/L=Raanana/O=Red Hat/CN=my CA)
```

spicec looks for `%APPDATA%spicecspice_truststore.pem` / `$HOME/.spicec/spice_truststore.pem`. This needs to be identical to the `ca-cert.pem` on the server, i.e. the ca used to sign the server certificate. The client will use this to authenticate the server.

### 3.24.6 Network is already in use

If running VMs crash with that message:

libvirt error code: 1, message: internal error: Network is already in use by interface

You are probably running Ravada inside a virtual machine or you are using the private network that KVM uses for another interface. This is likely to happen when running Ravada in a Nested Virtual environment.

**Solution: Change the KVM network definition. Edit the file `/etc/libvirt/qemu/networks/default.xml` and replace all the**

192.168.122 network instances by another one, ie: 192.168.123.

```
sudo virsh net-edit default
<ip address='192.168.122.1' netmask='255.255.255.0'>
  <dhcp>
    <range start='192.168.122.2' end='192.168.122.254' />
  </dhcp>
</ip>
```

Then reboot the whole system.

### 3.24.7 Copy & paste integration does not work

Make sure that the VM has a Spice communication channel (`com.redhat.spice.0`) and that the guest additions have been installed.

The Spice channel can be added through virt-manager's Add Hardware wizard or editing the XML:

```
<channel type='spicevmc'>
  <target type='virtio' name='com.redhat.spice.0' />
  <address type='virtio-serial' controller='0' bus='0' port='1' />
</channel>
```

Linux guests must install the `spice-vdagent` package, while Windows guests require [this installer](#) (source)



### 3.24.8 Resizing the viewer window does not change the guest display resolution

This feature requires the Spice communication channel and the guest additions. See above for instructions.

### 3.24.9 Windows 10 perfomance issues

thanks to @rlunardo

- Windows10 Enterprise ISO image (Home/Professional/Enterprise) before April 2017: if you install Enterprise version, it does not reach the end of installation. Issue posted on 30/10/2017. The Professional version does complete the installation. Recent Enterprise ISO image release completes the installation also.
- Windows 10 tuning after installation: There are several web site where we can find informations and solutions to solve CPU, RAM, Disk overload on Windows 10. Here some links:
  - <https://www.drivethelife.com/windows-10/fix-high-ram-cpu-memory-usage-after-windows-10-update.html>
  - <https://fossbytes.com/how-to-fix-high-ram-and-cpu-usage-of-windows-10-system-ntoskrnl-exe-process/>
  - <https://youtu.be/iHzEp8a8w10>

### 3.24.10 Problems with the time of the VM guest

You create a VM and you set the time correctly. After this VM becomes base and the time appears altered (-2h, +2h,...)

This is due to the parameter:

```
<clock offset='utc'> vs <clock offset='localtime'>
```

You can modify XML file from the command:

```
# virsh edit <machine_name>
```

### 3.24.11 Content is empty after upgrade

If after upgrade you get an almost blank screen it is probably a matter of cached javascript or CSS. Type SHIFT+F5 or SHIFT+Reload button in your browser. You could also clean browser history so it gets refreshed the next time you access the Ravada frontend.

### 3.24.12 Error: Web Service connection failed

If you get this error on the frontend, carefully review this documentation :

<https://ravada.readthedocs.io/en/latest/docs/apache.html>

It is very common to miss some step, so please take your time and restart all the services afterwards.

### 3.24.13 Virtual machines won't complete the shutdown

This may happen in some upgrades to Ubuntu 20.04. Virtual machines appear in the state *in shutdown* and won't complete the shutdown process.

You may notice if you type *virsh list*:

```
# virsh list
 Id   Name           State
-----
 2    tst20-04      in shutdown
```

The solution is to change the apparmor settings. Edit the file */etc/apparmor.d/usr.sbin.libvirtd* below:

```
# For communication/control from libvirtd
signal (receive) peer=libvirtd,
signal (receive) peer=/usr/sbin/libvirtd
```

Reboot the server and the virtual machines shutdown will work again.

*Thanks to Jim Fehlig from Suse.com*

### 3.24.14 internal error: cannot load AppArmor profile libvirt

Set *security\_driver = "none"* in */etc/libvirt/qemu.conf*

```
# security_driver = [ "selinux", "apparmor" ]
#security_driver = "selinux"
security_driver = "none"
```

Then restart libvirt service

```
$ sudo systemctl restart libvirtd
```

### 3.24.15 Cloned Ubuntu Server VMs assigned same IP

When cloning a virtual machine with Ubuntu Server (possibly applicable to other distributions as well), it has been observed that the cloned machines are assigned the same machine ID in the */etc/machine-id* file. As a result, the Netplan networking configuration attempts to resolve DHCP based on this ID instead of the MAC address, resulting in both machines being assigned the same IP address.

To resolve this issue, edit the file */etc/netplan/<configuration-file>.yaml* adding the next lines in the proper interface:

Add the following lines to the network configuration for each cloned machine:

```
dhcp4: yes
dhcp-identifier: mac
```

Apply the updated Netplan configuration:

```
$ sudo netplan apply
```

With these steps, each cloned virtual machine will be assigned a unique IP address based on its MAC address, resolving the issue of both machines receiving the same IP.

## 3.25 Upgrade Ravada

We try to make the upgrading procedure easy. Also if you are running a very ancient or legacy release, fear not ! We did our best so everything is set up with minimal user intervention.

In order to upgrade Ravada, you have to do a few steps:

### 3.25.1 Steps for a clean update

#### Step 1

Download the *deb* package of the new version found at the [UPC ETSETB repository](http://infoteleco.upc.edu/img/debian/ravada_2.2.1_ubuntu-20.04_all.deb).

```
$ wget http://infoteleco.upc.edu/img/debian/ravada_2.2.1_ubuntu-20.04_all.deb
```

#### Step 2

Install the *deb* package.

```
$ sudo apt install ./ravada_2.2.1_ubuntu-20.04_all.deb
```

On some upgrades may be required to install some dependencies. You will see because the packaging system will warn about it:

```
dpkg: dependency problems prevent configuration of ravada:
 ravada depends on libdatetime-perl; however:
 Package libdatetime-perl is not installed.
```

If so, install those dependencies automatically running:

```
$ sudo apt-get -f install
```

#### Step 3

Reconfigure systemd.

```
$ sudo systemctl daemon-reload
```

#### Step 4

Restart the services.

```
$ sudo systemctl restart rvd_back
$ sudo systemctl restart rvd_front
```

### Step 5

Check the apache configuration

If you upgrade from older releases you may have to add some lines to the apache proxy configuration. Check the [Apache proxy guide](#).

### Problems upgrading

Problems may arise please take a look at our [troubleshooting](#) guide. If everything fails you may [contact us](#) for assistance.

## 3.26 Upgrade Ravada with Rollback

This is the upgrade procedure when you want to keep everything just in case you wanted to roll back to the previous version.

### 3.26.1 Step 1: Shutdown the services

```
$ sudo systemctl stop rvd_back  
$ sudo systemctl stop rvd_front
```

### 3.26.2 Step 2: Keep the package and data

#### Step 2.1: Keep the package file

If you installed the package it must be in the server somewhere. It is a *deb* file called `ravada_x.y.z_system-version_all.deb`.

When in doubt, we keep most of the released packages in the [UPC ETSETB repository](#).

#### Step 2.2: Save Current Data

Check the database user and password from the config file `/etc/ravada.conf`. Then dump the database:

```
$ mysqldump -u rvd_user -p ravada > ravada.sql
```

### 3.26.3 Step 4: Upgrade

#### Step 4.1: Fetch the new package

Download the *deb* package of the new version found at the [UPC ETSETB repository](#).

```
$ wget http://infoteleco.upc.edu/img/debian/ravada_2.2.1_ubuntu-20.04_all.deb
```

### Step 4.2: Install

Install the *deb* package.

```
$ sudo apt install ./ravada_2.2.1_ubuntu-20.04_all.deb
```

On some upgrades may be required to install some dependencies. You will see because the packaging system will warn about it:

```
dpkg: dependency problems prevent configuration of ravada:
  ravada depends on libdatetime-perl; however:
  Package libdatetime-perl is not installed.
```

If so, install those dependencies automatically running:

```
$ sudo apt-get -f install
```

### Step 4.3: systemd

Reconfigure systemd.

```
$ sudo systemctl daemon-reload
```

### Step 4.4: apache config

Check the apache configuration

If you upgrade from older releases you may have to add some lines to the apache proxy configuration. Check the [Apache proxy guide](#).

### Step 4.5: start

Restart the services.

```
$ sudo systemctl restart rvd_back
$ sudo systemctl restart rvd_front
```

If you are upgrading from a very old release, it may take a while to proceed. You may check the log file for information opening another terminal:

```
$ sudo tail -f /var/log/syslog
```

Check the daemons are running:

```
$ sudo systemctl status rvd_back
$ sudo systemctl status rvd_front
```

Finally connect to your server and try to run and clone a virtual machine.

### 3.26.4 Rollback

If something failed and you wanted to rollback follow the [Rollback Ravada version guide](#)

## 3.27 Rollback Ravada Version

If you just upgraded Ravada and want to go to a previous version follow these steps:

### 3.27.1 Restore Database

This may not be necessary, try to install first the previous Ravada version and see if everything works. Anyway if you want to restore the database you must have a backup file.

```
$ mysql -u rvd_user -p ravada < ravada.sql
```

### 3.27.2 Install previous version

```
$ sudo dpkg -i ravada_version.deb
```

### 3.27.3 Restart the services

```
$ sudo systemctl restart rvd_back  
$ sudo systemctl restart rvd_front
```

## 3.28 SPICE client setup for Windows

### 3.28.1 Download Virt Viewer

Windows clients requires the [virt-viewer](#) tool to connect to their Virtual Machine.

### 3.28.2 Fix Windows registry

If *virt-viewer* won't start automatically after when viewing the virtual machine, add this to the Registry, or download [spice.reg](#). (Thanks to [@gmiranda](#)).

```
Windows Registry Editor Version 5.00
```

```
[HKEY_CLASSES_ROOT\spice]
```

```
@="URL:spice"
```

```
"URL Protocol"=""
```

```
[HKEY_CLASSES_ROOT\spice\DefaultIcon]
```

```
@="C:\\Program Files\\VirtViewer v5.0-256\\bin\\remote-viewer.exe,1"
```

```
[HKEY_CLASSES_ROOT\spice\Extensions]
```

(continues on next page)

(continued from previous page)

```
[HKEY_CLASSES_ROOT\spice\shell]
@="open"

[HKEY_CLASSES_ROOT\spice\shell\open]

[HKEY_CLASSES_ROOT\spice\shell\open\command]
@="\"C:\\Program Files\\VirtViewer v5.0-256\\bin\\remote-viewer.exe\" \"%1\""
```

## 3.29 SPICE client setup for MacOS

### 3.30 Virt-Viewer

If you don't have brew installed, visit [Homebrew](#).

Follow this steps:

1. Install a working (and compiled) version of **virt-viewer**. You may view the homebrew package's upstream source on [GitHub](#).

```
brew tap UPC/homebrew-virt-manager
brew install virt-manager virt-viewer
```

2. Once that's installed should be able make a call **remote-viewer** with a spice file, for example 405.vv file downloaded from Ravada.

```
remote-viewer 405.vv
```

### 3.31 Associate SPICE files with remote viewer

We want remote-viewer to automatically start and open the session when we double click the VM entry in Ravada. To do that we need to first create a small helper application.

1. Launch Automator and select Application from the dropdown list, when prompted.
2. Search for shell and drag to the right. The contents:

```
/usr/local/bin/remote-viewer "$@"
```

Make sure to select as arguments for passing the input. Save as **~/Applications/ravada-spice-launcher.app**

3. Locate a ravada spice file .vv file or any file with .vv extension, and then hold down the Control key. With the Control key pressed, click on the .vv file, and then right click, open with, look for the .app file you just made, and check the Always Open With checkbox in the bottom of the dialog. This took a couple of tries for it to stick, but eventually remembered.

In Chrome, click on the small arrow on the list of downloads at the bottom, and select "Always open files of this type" and select ravada-spice-launcher app.

If everything is set up correctly you should be able to double-click on the VM and remote-viewer should start up and take care of the rest.

That's all. Enjoy Ravada.

There is another way to do it if you have some troubles, install only RemoteViewer.

### 3.32 RemoteViewer

```
brew install --cask remoteviewer
```

#### Usage:

```
remote-viewer console.vv
```

Binary path in my env is /opt/homebrew/bin/remote-viewer.

Remember to allow this application in Settings -> Privacy & Security

You can see this message: "RemoteViewer.app" was blocked from use because it is not from an identified developer  
Enable the button: Open Anyway

### 3.33 Problems

1. If you have some trouble, check your remote-viewer path. Maybe it is different from /usr/local/bin.

```
which remote-viewer
```

Other path can be: /opt/homebrew/bin/remote-viewer

2. You have a similar repo installed.

```
Error: Formulae found in multiple taps:
```

Fix with:

```
brew untap jeffreywildman/virt-manager
```

### 3.34 SPICE client setup for MacOS

### 3.35 Virt-Viewer

If you don't have brew installed, visit [Homebrew](#).

Follow this steps:

1. Install a working (and compiled) version of [virt-viewer](#). You may view the homebrew package's upstream source on [GitHub](#).

```
brew tap UPC/homebrew-virt-manager  
brew install virt-manager virt-viewer
```

2. Once that's installed should be able make a call **remote-viewer** with a spice file, for example 405.vv file downloaded from Ravada.



```
remote-viewer 405.vv
```

### 3.36 Associate SPICE files with remote viewer

We want remote-viewer to automatically start and open the session when we double click the VM entry in Ravada. To do that we need to first create a small helper application.

1. Launch Automator and select Application from the dropdown list, when prompted.
2. Search for shell and drag to the right. The contents:

```
/usr/local/bin/remote-viewer "$@"
```

Make sure to select as arguments for passing the input. Save as **~/Applications/ravada-spice-launcher.app**

3. Locate a ravada spice file .vv file or any file with .vv extension, and then hold down the Control key. With the Control key pressed, click on the .vv file, and then right click, open with, look for the .app file you just made, and check the Always Open With checkbox in the bottom of the dialog. This took a couple of tries for it to stick, but eventually remembered.

In Chrome, click on the small arrow on the list of downloads at the bottom, and select “Always open files of this type” and select ravada-spice-launcher app.

If everything is set up correctly you should be able to double-click on the VM and remote-viewer should start up and take care of the rest.

That’s all. Enjoy Ravada.

There is another way to do it if you have some troubles, install only RemoteViewer.

### 3.37 RemoteViewer

```
brew install --cask remoteviewer
```

#### Usage:

```
remote-viewer console.vv
```

Binary path in my env is /opt/homebrew/bin/remote-viewer.

Remember to allow this application in Settings -> Privacy & Security

You can see this message: “RemoteViewer.app” was blocked from use because it is not from an identified developer  
Enable the button: Open Anyway

### 3.38 Problems

1. If you have some trouble, check your remote-viewer path. Maybe it is different from /usr/local/bin.

```
which remote-viewer
```

Other path can be: /opt/homebrew/bin/remote-viewer

2. You have a similar repo installed.

Error: Formulae found **in** multiple taps:

Fix with:

```
brew untap jeffreywildman/virt-manager
```

### 3.39 How to change the controller driver of a Windows VM to VirtIO

1. Add a new virtio disk (a small one, even 100mb will be enough. We will not be using it)
  - Use virt manager
  - or `virsh edit`
2. Boot the machine
3. Make sure Windows recognises the disk controller and has the drivers for it (the disk should be visible in `diskmgmt.msc`)
4. Set boot to fail safe mode
  - Launch an elevated command prompt
  - Type `bcdedit /set {current} safeboot minimal`
5. Shut down
6. Change the main disk to virtio
7. Boot. Now get into the admin prompt and disable failsafe mode.
  - Launch an elevated command prompt
  - Type `bcdedit /deletevalue {current} safeboot`
8. Reboot and make sure everything works.
9. Now you can shut down the machine and remove the small virtio disk

Source: <http://triplescomputers.com/blog/uncategorized/solution-switch-windows-10-from-raidide-to-ahci-operation/>

### 3.40 Backup Virtual Machines

Since release 1.5 it is possible to back up and restore Virtual Machines. This feature can be used to recover deleted machines, restore contents after changes and even to migrate machines to another server.

The backup and restore procedures must be used from the command line.

### 3.40.1 Synopsis

```
$ sudo rvd_back --backup machine1
$ sudo rvd_back --restore /var/lib/libvirt/images/backup/machine1-2022-04-06.tgz
```

### 3.40.2 Backup

Perform the backup calling the *rvd\_back* tool with the names of all the virtual machines you want to backup.

```
$ sudo rvd_back --backup machine1 machine2 ... machineN
```

#### Clones and bases

Only the files and definitions are stored in the backup file. If you want to backup a clone and you want to make sure you can restore it correctly, do a backup of the parent base too.

```
$ sudo rvd_back --backup base clone1 clone2 ...
```

#### Owner

Information and metadata of the virtual machine is backed up too. If you ever removed the owner but you restore the virtual machine later, the user information such as name and password will be restored too.

### 3.40.3 Restore

To restore a Virtual Machine you must pass the path of the filename where the compressed backup has been stored.

```
$ sudo rvd_back --restore /var/lib/libvirt/images/backup/machine1-2022-04-06.tgz
```

If Ravada detects this virtual machine is already there it will ask for confirmation. Metadata and user information will be restored along with filesystem contents.

### 3.40.4 Migration to another server

The backup file can be used to migrate the virtual machine to another Ravada server. Just copy the file and call *rvd\_back --restore*. Be careful: if you want to migrate a clone, the base must be migrated first.

Owner users will be created in the new virtual machine if they don't exist. The users will be assigned regular user permissions. Specially granted privileges or administration roles must be set up manually after restoring.

```
$ sudo rvd_back --restore /home/machine1-2022-04-06.tgz
```

### 3.40.5 Backup directory

The default backup directory will be the folder *backup* inside the default storage pool. In most systems that will be located at: */var/lib/libvirt/images/backup*. You can set another directory from the *Node Management* menu in the frontend: go to *Admin Tools* and select *Nodes*, then in *KVM\_localhost* change the backup directory and save it.

options

bases

remove

name

host

base storage

clone storage

default storage

backup directory

KVM\_localhost

localhost

pool2

default

/var/lib/libvirt/images/backup

cancel

save

## 3.41 Virtual Machine Manual Migration

If you have several Ravada servers you may want to copy a virtual machine from one to another.

**Warning:** The easiest way to migrate Virtual Machines is using the new *Backup Virtual Machines* tool added in release 1.5.

In this example we use the old manual procedure to copy the base for a virtual machine called *Lubuntu-1704*.

### 3.41.1 Check the storage pools directories

If the filesystem layout of both servers is different, the migration can be complicated. You have to bear in mind the disk volumes may have backing files in a different location and you need to rebase it.

Check first the directory where the volumes are stored, if you can't have them in both servers you have to create a clone, spin if off and migrate it to the new server.

In this example we inspect first the full path of the volumes.

```
root@origin:~# virsh dumpxml Lubuntu1704 | grep "source file"

<source file='/var/lib/libvirt/images/lubuntu-vda-k1dj.qcow2' />
```

For each source file, check its backing file:

```
root@origin:~# qemu-img info /var/lib/libvirt/images/lubuntu-vda-k1dj.qcow2 | grep
↪ "backing file"

backing file: /var/lib/libvirt/images.2/lubuntu-vda.ro.qcow2
```

So we now know we have to move one file to `/var/lib/libvirt/images/` and another one to `/var/lib/libvirt/images.2/`. Check if the new server has these directories in storage pools.

```
root@destination:~# virsh pool-list
```

```

Name           State    Autostart
-----
default active yes
```

```
root@destination:~# virsh pool-dumpxml default | grep path
<path>/var/lib/libvirt/images</path>
```

In this case migrate the virtual machine as it is will be difficult. We have to rebase the volumes. The easiest way would be to create a clone, spin off the volumes from the web frontend administration and migrate it then.

### 3.41.2 Import the Base

#### Copy the Base definition

First copy the base definition file from server origin to destination. You need an user in the destination machine and ssh connection from each other.

```
root@origin:~# virsh dumpxml Lubuntu1704 > Lubuntu1704.xml
root@origin:~# scp Lubuntu1704.xml root@dst.domain:
```

#### Copy the volumes

The volumes have a backing file, you must find out what it is so you can copy to destination.

```

root@origin:~# grep source Lubuntu1704.xml
<source file='/var/lib/libvirt/images/Lubuntu1704-vda-X18J.img' />
root@origin:~# qemu-img info /var/lib/libvirt/images/base-vda-X18J.img | grep -i backing
backing file: /var/lib/libvirt/images/Lubuntu1704-vda-X18J.ro.qcow2
root@origin:~# rsync -avPS /var/lib/libvirt/images/base-vda-X18J.ro.qcow2 root@dst.
↪ domain:/var/lib/libvirt/images
root@origin:~# rsync -avPS /var/lib/libvirt/images/Lubuntu1704-vda-X18J.img root@dst.
↪ domain:/var/lib/libvirt/images
```

#### Define the base on destination

Go to the destination server and define the virtual machine base with the XML config you copied before

```
root@dst:~# virsh define Lubuntu1704.xml
Domain base defined from Lubuntu1704.xml
```

### Import the base to Ravada on destination

Run this command and you should see the base on the Ravada web admin page.

```
root@dst:~# rvd_back --import-domain Lubuntu1704
This virtual machine has 3 backing files. Do you want to import it as a base ? Please_
↪answer y/n [yes]:
```

### 3.41.3 Importing clones

Now if you want to import a clone too, first you have to ask the clone owner to start and stop the machine on destination. Then you have to copy the volumes from origin and overwrite what has just been created on destination.

#### Create a clone

The owner of the original clone must create a clone in destination using Ravada. That will create a basic virtual machine with the same name owned by the correct user. Stop the domain on destination:

```
root@dst:~# virsh shutdown Lubuntu1704-juan-ramon
```

Make sure it is stopped

```
root@dst:~# virsh dominfo Lubuntu1704-juan-ramon
```

#### Copy the clone volumes

Find out what are the clone volume files, and copy them to the temporary space in destination:

```
root@origin:~# virsh dumpxml Lubuntu1704-juan-ramon | grep "source file" | grep -v ".ro."
<source file='/var/lib/libvirt/images/Lubuntu1704-juan-ramon-vda-kg.qcow2' />
root@origin:~# rsync -av /var/lib/libvirt/images/Lubuntu1704-juan-ramon-vda-kg.qcow2_
↪root@dst:/var/lib/libvirt/images
```

#### Start the clone on destination

First move the volumes to the right place, notice in destination the volumes have different names. Check the XML configuration matches the place where you stored the qcow files.

Hopefully then you can start the clone. It is a delicate procedure that must be followed carefully, please consider helping with this document if you have any suggestions.

### 3.41.4 Importing Standalone Machine

#### Dumping data

To export the necessary data you need the XML file and the volume files.

#### XML file

```
root@src:~# virsh dumpxml Lubuntu1704 > Lubuntu1704.xml
```

#### Volume Files

Find out the volume files searching for “source file” in the XML. Copy these files to the destination server.

```
root@origin:~# virsh dumpxml Lubuntu1704 | grep "source file" | grep -v ".ro."
<source file='/var/lib/libvirt/images/Lubuntu1704-vda-kq.qcow2'/>
root@origin:~# rsync -av /var/lib/libvirt/images/Lubuntu1704-vda-kq.qcow2 root@dst:/var/
↪ lib/libvirt/images/
```

#### Importing data

First of all you need to check the storage directory in the destination server matches the source. Check for “source file” lines in the XML and change it if you will place the qcow files elsewhere.

Then define the virtual machine base with that XML config file.

```
root@dst:~# virsh define Lubuntu1704.xml
Domain base defined from Lubuntu1704.xml
```

Then try to start the virtual machine:

```
root@dst~# virsh start Lubuntu1704
```

Once you have verified the machine is running, import it into Ravada:

```
root@dst~# rvd_back --import-domain Lubuntu1704
```

## 3.42 Kiosk Mode

This feature is available since Ravada version 1.1 currently in development due April 2021. Download it from <http://infoteleco.upc.edu/img/debian/>

Kiosk ( or anonymous ) allows any user, not logged in, to create a volatile virtual machine. Once this machine is shut down, it is destroyed automatically.

### 3.42.1 Setting

This *kiosk* mode must be defined for some bases in some networks.

### 3.42.2 Define a Network

You can allow kiosk mode from any network, but you can define a new network where this mode is allowed.

Go to Admin Tools, Networks and click *New Network*.

Fig. 1: New network form

Set a name for this network definition, you may want to allow access without client password. Enable *all machines* so users from this network can access the virtual machines.

### 3.42.3 Allow anonymous mode

You can allow anonymous access to some bases from the networks management form.

Go to Admin Tools, Networks and click in the name of the network. If you want to grant anonymous users to everyone use the *default* network. This is not a good practice unless the server is behind a firewall. Otherwise create a new network and grant anonymous access only to the users that come from there.

Click on the button *Machines*. A list of all the bases will be shown. For a base to be used anonymously it must be defined as public and the *anonymous* option must be selected.

In this example we configure anonymous access to the base called *focal*.

Public	Anonymous	Base
<input type="checkbox"/>	<input type="checkbox"/>	aa-1
<input checked="" type="checkbox"/>	<input type="checkbox"/>	alpine
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	focal
<input type="checkbox"/>	<input type="checkbox"/>	focal-2020
<input type="checkbox"/>	<input type="checkbox"/>	tstbuster
<input type="checkbox"/>	<input type="checkbox"/>	ztest
<input type="checkbox"/>	<input type="checkbox"/>	zz-test-base
<input checked="" type="checkbox"/>	<input type="checkbox"/>	zz-test-base-alpine



Setting a base as *public* allows any known user with access to run it. Anonymous gives access to everyone in your network without an username.

### 3.42.4 Auto remove anonymous machines

Virtual machines created for anonymous users can be easily and automatically removed. Go to Admin Tools, Machines, and click in the name of the base. In the *Options* tab enter a *timeout* and optionally *shutdown disconnected*.

In this example the virtual machine will be destroyed when the user closes the viewer or after running for 60 minutes.

## Virtual Machine focal Settings

Description	Max memory (MB)	1024	
Rename	Current memory (MB)	1024	
Options	CPUs	1	
Drivers	Timeout	60	The machine will shutdown after these minutes
Hardware	Volatile Clones	<input type="checkbox"/>	Clones created from this machine will be removed on shutdown.
Graphics	Shutdown disconnected	<input checked="" type="checkbox"/>	Virtual Machine will be shut down when user disconnects.
Base			

### 3.42.5 Access

Access now to the anonymous section in your ravada web server. <http://your.server.com/anonymous>

You should see there the base of the virtual machine you allowed before.

## 3.43 How to do a volatile Virtual Machine

This means that the virtual machine will be removed when shutdown.

**Note:** If you enable this option in a base machine, the clones that are created thereafter will be volatile. The VMs cloned so far, if they existed, are not affected.

### 3.43.1 Enable/disable this option

You can enable or disable this option from the VM settings in the options tab.

Virtual Machine **centos-admin-admin** Settings

Description
Rename
Options
Drivers
Hardware
Screenshot
Graphics
Copy
Remove Machine

Max memory (MB)
2000
Current memory (MB)
2000
Timeout
The machine will shutdown after these minutes
Volatile Clones
☒
Clones created from this machine will be removed on shutdown.
Owner
admin
Change the owner of the machine
Trameta la consulta
Cancel

## 3.44 Ravada Cluster

### 3.44.1 Adding support for multiple Ravada nodes

#### Description

From release 0.4 we have clustering support for Ravada. This allows the administrator to add more physical hosts so virtual machines are balanced and distributed among all them.

lubuntu	✓	☑	Down	This Machine is a base				
<i>lubuntu-anon48ba</i> [ Cloned ]	Volatile	<input type="checkbox"/>	<div>Running</div> <div>disconnected</div>				nenufar	
<i>lubuntu-anon48dr</i> [ Cloned ]	Volatile	<input type="checkbox"/>	<div>Running</div> <div>disconnected</div>				hortensia	
<i>lubuntu-anon48ee</i> [ Cloned ]	Volatile	<input type="checkbox"/>	<div>Running</div> <div>disconnected</div>				hortensia	
<i>lubuntu-anon48jq</i> [ Cloned ]	Volatile	<input type="checkbox"/>	<div>Running</div> <div>disconnected</div>				nenufar	
<i>lubuntu-anon48rm</i> [ Cloned ]	Volatile	<input type="checkbox"/>	<div>Running</div> <div>disconnected</div>				hortensia	
<i>lubuntu-anon48ze</i> [ Cloned ]	Volatile	<input type="checkbox"/>	<div>Running</div> <div>disconnected</div>				hortensia	
<i>lubuntu-f</i> [ Cloned ]	<input type="checkbox"/>	<input type="checkbox"/>	<div>Running</div> <div>disconnected</div>					localhost

This feature is currently stable. It works but there are some management forms that we have to polish, and we need to test it more.

## Nodes

A node is a Ravada backend server that runs the virtual machines. With the current implementation it is required a main master node and secondary nodes can be added later. The master node, also known as local node should be a rock solid server. It will also run some of the virtual machines as they get balanced among all the nodes. Having fault tolerance master node is beyond the scope of this document, but we expect to add a new document for examples of how to achieve it in the future. Contributions welcome.

Secondary nodes are disposable physical hosts. It is our goal to allow these nodes be up and down at the administrator will and that they could be easily added to a cluster. Ravada should be able to cope with sudden death of those nodes, though it is better if they are disabled and shut down nicely from the Ravada node administration frontend.

---

**Note:** A Base virtual machine must always exist first in the main host. Then you can enable it in some secondary nodes. It is not possible to have a base in a node but not in the main server. If you remove the base from main it will become a normal virtual machine.

---

## Storage

Shared storage is not necessary. The base volumes will be synced across the nodes by Ravada itself. When bases are created, the administrator can configure on which nodes will be available for the virtual machines.

All the master and nodes must have the same storage pools configuration.

## New node

### Requirements

To add a new node to a Ravada cluster you have to install a minimal Linux operating system with these packages. Some distributions have different package names.

- openssh-server
- libvirt-bin or libvirt-daemon
- libvirt-daemon-system
- libvirt-clients
- qemu-kvm
- rsync

It is possible to have nodes with heterogeneous operative systems: different Ubuntu, Debians, even RedHats or Fedoras can be added, though it should be easier if all of them are similar or identical if possible.

In some operative systems like Debian, you must create a link to the kvm file to get the node correctly working. It is not necessary for Ubuntu or Fedora/RedHat.

```
$ ln -s /usr/bin/kvm /usr/bin/kvm-spice
```

### Configuration

The master ravada node needs to access to the secondary nodes through SSH. So password-less access must be configured between them. This is an example of configuring in Debian and Ubuntu servers. Other flavours of linux should be quite the same.

#### Before you start

Most of the usual mistakes come from not running the commands from the root user. Follow the guide carefully and double check you are following the instructions exactly. When you see a code like this it means it must run from the root user in the master node:

```
root@master:~#
```

This means you must run as the root user in the node.

```
root@node:~#
```

To become root user you must either login as such or run sudo to become root:

```
frankie@master:~ sudo bash
```

#### Configure Node

First, temporary allow *root* access with *ssh* to the remote node.

```
root@node:~# vi /etc/ssh/sshd_config
PermitRootLogin yes
```

Then set a root password and restart ssh service. Warning: only do this if you haven't already set a password for the root user in this host. If you are using an Ubuntu server you probably haven't, but debian based distributions usually set the root user on install.

```
root@node:~# passwd
Enter new UNIX password: *****
root@node:~# systemctl restart ssh
```

#### Configure Master

Check you can access with *root* from master to node:

```
root@master:~# ssh node
```

You may already have a public/private key created in the master node. Check if there are *id\*pub* files in */root/.ssh* directory. Create the keys otherwise:

```
frankie@master:~ sudo bash
root@master:~# ls /root/.ssh/id*pub || ssh-keygen
```

Now you must copy the public ssh key from master to node:

```
root@master:~# ssh-copy-id -i /root/.ssh/id_rsa.pub node
```

Check it works:

```
root@master:~# ssh node
```

Now you can restore the *PermitRootLogin* entry to the former state in the file */etc/ssh/sshd\_config* at *node*.

```
PermitRootLogin prohibit-password
```

## Security

It is advisable have a firewall configured in the node. Access restrictions should be enforced carefully. Only allow ssh login from the master server and other operational hosts from your network.

## Networking and Firewall

For the master node to start the other nodes it must have some open ports.

### Master

Master must be able to send packets on udp port 9 and tcp and udp port 7

```
# wake on lan
-A OUTPUT -p udp -m udp --dport 7 -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 7 -j ACCEPT
-A OUTPUT -p udp -m udp --dport 9 -j ACCEPT
```

### Nodes

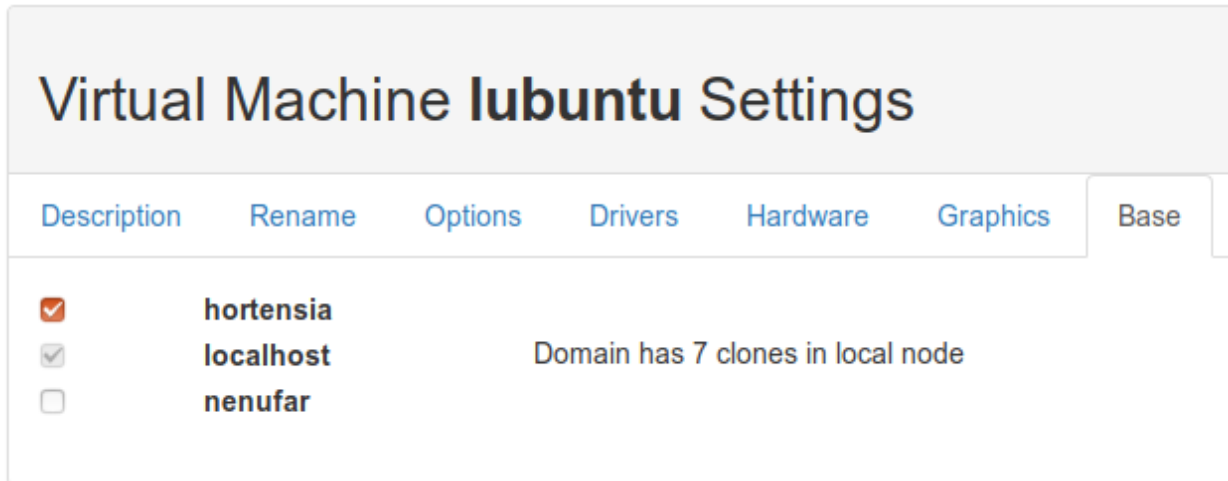
Nodes must accept packets on udp port 9 and tcp and udp port 7

```
# wake on lan
-A INPUT -p udp -m udp --dport 7 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 7 -j ACCEPT
-A INPUT -p udp -m udp --dport 9 -j ACCEPT
```

## Operation

Add nodes in the new section *Admin Tools - Nodes*

Allow a base to create clones in nodes checking them in the machine management section, at the *Base* tab.



Now try to create multiple clones from a base, they should get balanced among all the nodes including the master one.

### TroubleShooting

libvirt error code: 38, message: End of file while reading data: nc: unix connect failed: No such file or directory

It means you didn't install libvirt in the node. Fix it this way:

For Ubuntu 18.04:

```
root@node:~# apt-get install libvirt-bin
```

From Ubuntu 18.10 instead of libvirt-bin you need install libvirt-daemon-system and libvirt-clients.

```
root@node:~# apt-get install libvirt-daemon-system
root@node:~# apt-get install libvirt-clients
```

### Balance algorithm

How does Ravada decides which node will handle the next virtual machine ?

We may choose a different algorithm in the future, or even allow a setting to change it. Right now it goes that way:

The list of nodes gets sorted by the number of virtual machines running. If there is a tie, the node with more free memory is chosen.

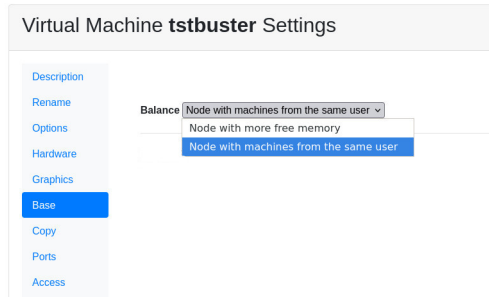
We have an extra step for speeding things up, when checking the nodes status, if one has less than 2 virtual machines running, and it has free available memory, it is picked.

There is a drawback we have yet to fix: if many virtual machines are started at the same time, some may probably go to the same node. This is because a lot of work is done in parallel.

## Selection algorithm

Now you can choose to run virtual machines in the node where the user has other machines started. This may be usefull if you want to do networking among virtual machines like in teaching environments.

Go to the base virtual machines settings and select the *Base* tab. Change the algorithm for all the bases you want to have started in the same node.



## TODO

We already know we have to improve:

- administration forms in the web front
- check if nodes storage gets filled
- documentation

This is a new feature, we are currently testing. Feedback welcome through our Telegram public forum <http://t.me/ravadavdi> or [github issues](#).

## 3.45 How to make a virtual machine disk sparse

When someone had deleted files to reduce the virtual machine img size, you need to do some actions in the server to return this free space to the server.

### 3.45.1 Steps

1. Install libguestfs-tools

```
$ apt install libguestfs-tools
```

2. Check the real size of the virtual machine size

```
$ qemu-img info file.qcow2
```

The output will be something that contains this information:

```
disk size: 10G
```

3. Make a backup copy of the img file

```
$ cp -p file.qcow2 /another/directory/file.backup.qcow2
```

4. Now use virt-sparsify

```
$ virt-sparsify --in-place file.qcow2
```

5. Check if the virtual img size has been reduced

```
$ qemu-img info file.qcow2
```

The output now shows that the size has decreased:

```
disk size: 5G
```

6. Check if the virtual machine works.
7. If the virtual machine works, then remove the img file backup.

Extended information [about](#) to reduce image size.

## 3.46 Server Hardware

So you want to buy a brand new server to run virtual machines on it. Here are some advices to achieve maximum performance and how to save some money.

### 3.46.1 Shared Storage

If you are planning to build a large cluster of Ravada nodes you may go to an expensive shared storage infrastructure. Be aware this can be a big deployment that can give access to thousands of users, but it will be expensive and you may have some performance issues.

See this doc about [Clustering Hardware recommendations](#)

Ravada is easy to grow, so try first a single server setup and you can add more later.

### 3.46.2 Initial Hardware Setup

#### Memory

RAM is the main issue. Multiply the number of concurrent workstations by the amount of memory each one requires and that is the total RAM the server must have. However, recent virtualization improvements allow you to overcommit the memory.

#### Network

For remote virtual desktops it is enough to have 1 GB ethernet cards. If you are planning on having many video intensive workstations at the same time it would be good to have 10 GB network cards on the host.

#### Disk Drives

You will need to store 3 different kind of data in the server:

- Operative System
- Bases volumes
- Clones volumes



## Operative System Disk drives

The Operative System partitions like root, /usr and /var are critical so the server keeps running. But performance is not the main issue. Buy two or three small hard disk drives. Create RAID1 or RAID5 and define these partitions there. In our experience RAID1 is more than enough, if you can afford it, buy 3 disk drives: 2 in the RAID and the third one as a spare.

There is no much space requirement for the operative system. 50 GB should be than enough. If you buy larger disks you may create a partition to store some virtual machines volumes.

## Bases Volumes

With Ravada virtual machines usually are cloned from a **base**. This base is prepared in advance with all the software the users need. All the machines will read information from the base volumes so it is a good idea to store this data in SSD disk drives.

This kind of disks are expensive, so you likely would want to buy only one or two small disk drives for the base volumes.

RAID5 is usually slow so it is not adviced. If you want to have redundancy configure a RAID1 with 2 SSD disk drives for the base volumes. It would usually be mounted at /var/lib/libvirt/images

If you want to save some money do not use RAID for the base volumes. In our experience, top hardware vendor brand disk drives are reliable. You may get more space if you buy 2 SSD drives and create two different partitions. In this case you will have base volumes stored in /var/lib/libvirt/images.1 and /va/lib/libvirt/images.2 .

Be aware that without RAID there is a downtime risk. If one of the disk drives fail the information it contains may be lost and a backup must be restored in a new replacement drive. But it is uncommon to have both disk drives failed at once, so you can restore the data in the other volume and carry on while the replacement arrives.

## Clones Volumes

Clones volumes are incremental information stored on top of base volumes. Usually this data doesn't require as much performance. So it is not a bad idea to save some money here and store the clones volumes in large mechanical disk drives. Anyway if you can afford it buy fast disks for a better user experience.

## Configuration Examples

Really cheap server

- Operative System: 2 x Hard Disk Drives 100 GB in RAID1
- Volumes: 1 x Solid State Disk drives

Budget performance server

- Operative System: 2 x HDD 100 GB in RAID1
- Bases Volumes: 1 x SSD 500 GB
- Clones Volumes: 1 x HDD 1 TB

From this example you can grow as long as your budget allows it. Having more drives may give you more space. If you need high availability 24x7 you have to duplicate the volumes disk drives and set RAID1.

High Availability and performance server

- Operative System: 2 x HDD 100 GB in RAID1

- Bases Volumes: 2 x SSD 500 GB in RAID1
- Clones Volumes: 2 x HDD 1 TB in RAID1

### 3.46.3 Growing and Scaling

Ravada is easy to grow. Start with a tight budget, but try to buy the faster drives you can afford.

When more users start virtual machines at the same time the server may run out of memory. Adding more RAM will give you more concurrent users.

If you run out of disk space you can buy more disk drives and add more storage pools. You can configure some bases go to one storage, others to another. If you buy more storage it can be defined the new clones will be created in new partitions.

**Ravada does scale:** A cluster can be created from a main server and nodes can be added to it. You can use older hardware, even PCs. Ravada will automatically balance virtual machines and start in the less used nodes. You don't need shared storage in the clusters, but if you use it start up and clone times will be much faster.

### 3.46.4 Backup

Borg backup is a good free choice, its main advantage is it has good deduplication features.

## 3.47 Cluster Hardware

### 3.47.1 Shared Storage

If you are planning to build a large cluster of Ravada nodes you may go to an expensive shared storage infrastructure. Be aware this can be a big deployment that can give access to thousands of users, but it will be expensive and you may have some performance issues.

Ravada is easy to grow, so try first a single [Server](#) setup and you can add more later.

### 3.47.2 Recommendations

This document needs more detailed information, for starters:

- Shared Storage with 10 GB or Fiber networking
- All Flash or mixed storage
- Two network switches for data channels

## 3.48 Ravada CLI

There are some things you can do from the CLI with Ravada.

This document is a work in progress. If you are interested in documenting more any feature [let us know](#).

### 3.48.1 Help

```
$ sudo rvd_back --help
```

### 3.48.2 LDAP

You can execute some LDAP actions from the command line.

#### Test LDAP connection

If you wonder if Ravada is able to access correctly to your LDAP server use the `-test-ldap` flag. First it will try to connect, then you can type an username and password to confirm it is a valid user.

```
$ sudo rvd_back --test-ldap
Connection to LDAP ok
login: jimmy.mcnulty
password: whatever
LOGIN OK bind
```

#### Create LDAP user

Add a new entry in your LDAP server. Warning the password will be shown in the clear.

```
$ $ sudo rvd_back --add-user-ldap jimmy.mcnulty
```

#### Create LDAP group

Add a new group in your LDAP server. These are POSIX groups with member uids inside.

```
$ $ sudo rvd_back --add-group-ldap staff
```

#### Add users to LDAP groups

Once you have users and groups in your LDAP server you can easily add member entries to a group. *Warning* : the user must have logged in at least once.

A list of known LDAP groups will be shown. If the user is already member of a group it will be flagged with a *YES*. Type the name of the new group you want the user to belong to:

```
$ $ rvd_back --add-user-group jimmy.mcnulty
$ - staff :
$ - cops : YES
$ - students :
$ - teachers :
$ Add user to LDAP group: teachers
```

### 3.48.3 Daily Hibernating and Stopping

With the CLI you can set automatic hibernation and shutdown of idle virtual machines. Check the documentation about [Automatic Daily Operations](#).

## 3.49 Automatic Daily Operations

It is possible to configure automatic daily cleaning operations. You may want to keep the system from having idle virtual machines. Using the *Ravada CLI* you can stop or hibernate machines at a given time.

### 3.49.1 Cron

We use the *cron* utility to execute operations at a given time. If you are not familiar with this consult the system documentation typing *man cron* or any other online manual.

To configure the cron entries type this from the host console:

```
$ sudo crontab -e
```

### 3.49.2 Usage

The most usual operation is to hibernate or shutdown all the inactive virtual machines at night. All the users that have an active connection will not be affected.

Also, any virtual machine marked as *auto start* will be kept running.

### 3.49.3 Examples

#### Hibernate disconnected

This cron entry will hibernate all virtual machines that have disconnected the remote viewer. This will be executed at 4 AM in the morning each day.

```
00 04 * * * /usr/sbin/rvd_back --hibernate --disconnected
```

#### Hibernate active

That will hibernate any active virtual machine. This will be executed at 4 AM in the morning in weekdays.

```
00 04 * * mon-fri /usr/sbin/rvd_back --hibernate --active
```

## Shutdown or Hibernate

You can use shutdown in the previous examples instead of hibernate. The main difference is hibernated machines must dump all the memory to disk and use large amounts of space in the server.

### 3.49.4 Other usage

The commands that can be issued are:

- shutdown
- hibernate

The modifiers to list virtual machines are:

- active
- disconnected
- all

## 3.50 Single Sign On

In this doc we show an example of how to configure the server to use a Single Sign On or CAS login service to login into Ravada.

In order to authenticate users into Ravada using an external Single Sign On service you need to configure the URL of the external service into ravada.conf YAML file.

### 3.50.1 Configuration

The config file usually is /etc/ravada.conf. Add this configuration:

```

sso:
  url: https://cas.example.com           # External CAS Service location
  service: http://ravada.example.com/login # Ravada Service URL
  logout: true                          # Set to also logout from CAS
  ↪Service when logout from Ravada
  cookie:
    priv_key: /etc/ravada/cas/priv_key.pem # Pathname of the private key of the
    ↪certificate that we will use to generate / validate session cookies
    pub_key: /etc/ravada/cas/pub_key.pem   # Pathname of the public key of the
    ↪certificate that we will use to generate / validate session cookies
    type: rsa                             # Signature algorithm:
    ↪dsa, rsa,...
    timeout: 36000                        # Session cookie lifetime
    ↪(in seconds

```

### 3.50.2 External libraries

You will need the `Authen::ModAuthPubTkt` library that can be downloaded from CPAN

```
$ sudo cpanm --force Authen::ModAuthPubTkt
```

### 3.50.3 Cookie key certificate

A key certificate must be created in the Ravada server to verify the SSO ticket. Use the same directory configured in the previous step. Also the webserver user must be able to read it. The ravada frontend service ( `rvd_front` ) is supposed to run as the user *ravada* , group *ravada*. We allow the group to read the contents of the key file this way:

```
$ sudo bash
$ cd /etc
$ mkdir ravada
$ chgrp ravada ravada
$ chmod g+rx ravada
$ chmod o-rx ravada
$ cd ravada
$ mkdir cas
$ chgrp ravada cas
$ chmod g+rx cas
$ chmod o-rx cas
$ cd cas
$ openssl genrsa -out priv_key.pem 2048
$ openssl rsa -in priv_key.pem -pubout -out pub_key.pem
```

Reset the service to enable the new settings:

```
$ sudo systemctl restart rvd_front
```

### 3.50.4 How it works

When an unauthenticated connection to Ravada is detected, the user is redirected to the CAS service authentication URL.

The CAS service authenticates the user and jumps to the Ravada service address by sending a session ticket as a parameter.

Ravada authenticates the ticket, obtains the user code and generates a session cookie for future user connections (since the ticket can only be authenticated once).

## 3.51 AppArmor troubles

### 3.51.1 Symptoms

If your machines hang while shutting down, this might be caused by AppArmor. This issue seems to appear only in Ubuntu systems upgraded to 20.04.

### 3.51.2 Possible workaround

<https://kifarunix.com/how-to-fix-qemu-kvm-not-connected-error-on-ubuntu-20-04/>

## 3.52 Service Status

Health and monitoring data can be accessed from /status.json

### 3.52.1 Configuration

Add this entry to /etc/rvd\_front.conf

```
,status => {  
  allowed => [  
    '127.0.0.1'  
    , 'ip1.domain'  
    , 'ip2.domain'  
  ]  
}
```

## 3.53 Adding Custom Messages

You can add a custom header and text at the login screen.

### 3.53.1 Configuration

Configure the file rvd\_front.conf like this:

```
{  
  login_header => 'Login'  
  , login_message => 'Login to Start a Machine'  
}
```

## 3.54 Create a custom login template

If you need custom login template create one and save it in /usr/share/ravada/templates/main/custom, e.g. custom\\_login.html.ep

### 3.54.1 Custom login template contents

The default custom file can be found at `/usr/share/ravada/templates/main/start.html.ep`. Use it as a guide for your own template.

Be aware:

- form must be `method=post`
- keep the login input type `name=login`
- keep the password input type `name=password`
- keep the hidden input with the `name=url` entry
- the submit button must be called `name=submit`

### 3.54.2 Configuration

Add your template in `/etc/rvd_front.conf`

**Warning:** Check that `rvd_front.conf` exists. If you work on a Development release you have an example here `etc/rvd_front.conf.example`.

**Warning:** Do not include the extension file `.html.ep` in the path. E.g. `custom_login.html.ep` -> `custom_login`

`,login_custom => 'main/custom/custom_login'`

### 3.54.3 Path for CSS, js and images

The custom files must be placed in `/usr/share/ravada/templates/main/custom`

If CSS, js or images are needed save in: `public/css/custom`, `public/js/custom` or `public/img/custom` respectively. These files must be located inside `/usr/share/ravada/templates/public`.

---

**Note:** Make sure your CSS, JS or images in custom template refers to those paths.

---

### 3.54.4 Restart frontend

Finally restart `rvd_front`:

```
$ sudo systemctl restart rvd_front
```



## 3.55 Create a custom footer template

If you need custom footer template create one and save it in `/usr/share/ravada/templates/main/custom/`, e.g. `custom/custom_footer.html.ep`

### 3.55.1 Configuration

Add your template in `/etc/rvd_front.conf`

**Warning:** Check that `rvd_front.conf` exists. If you work on a Development release you have an example here `/etc/rvd_front.conf.example`.

**Warning:** Do not include the extension file `.html.ep` in the path. E.g. `custom_footer.html.ep -> custom_footer`

```
,footer => 'main/custom/custom_footer'
```

### 3.55.2 Path for CSS, js and images

If CSS, js or images are needed save in: `public/css/custom`, `public/js/custom` or `public/img/custom` respectively.

---

**Note:** Make sure your CSS, JS or images in custom template refers to those paths.

---

### 3.55.3 Restart frontend

Finally restart `rvd_front`:

```
$ sudo systemctl restart rvd_front
```

## 3.56 Disable Spice Password

When the users start a virtual machine, a password is defined for the spice connection. This can be disabled for a given network or in the main settings.

If you want to disable Spice passwords for all the virtual machines in all the networks, disable it in *Admin Tools*, then *Settings*. The option is *Display Password*.

Follow the next guide if you want to disable the display password only for clients that access from some networks.

### 3.56.1 Define the network

Choose the *Admin Tools* option *Networks*. Create a new network and enable the settings that suit you best.

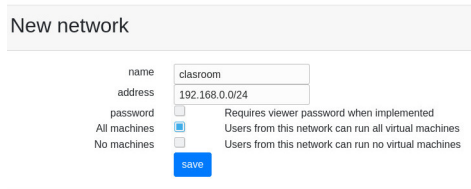


Fig. 2: New network form

Set a name for this network definition, you may want to allow access without client password. Enable *all machines* so users from this network can access the virtual machines.

### 3.56.2 Applying settings

This settings applies on starting a new virtual machine. So running virtual machines will keep the former settings. Shutting them down and up will trigger the new configuration.

### 3.56.3 Default setting

Any other network requires password as defined by the '0.0.0.0/0' network setting.

### 3.56.4 Why is that ?

Ravada opens SPICE connections and manages iptables to make sure no one can connect to another user's virtual machine. This is also enforced through the password setting. Please consider disabling it only in controlled, seat-unique ip environments. Configuring SPICE-TLS is recommended.

## 3.57 Ravada advanced settings

### 3.57.1 Display IP

On a server with many IP address, the configuration file allows the administrator define which one is used for the display. Change the field *display\_ip* in the Virtual Managers (VMs) table. Set there the public address of the server.

---

**Tip:** With recent releases of ravada this shouldn't be necessary. Ravada uses the routing table from the server and sets the display IP from there.

---

If you want to force the display IP do it this way:

For Ravada 0.9 and older set it at `/etc/ravada.conf`:

```
display_ip: public.display.ip
```

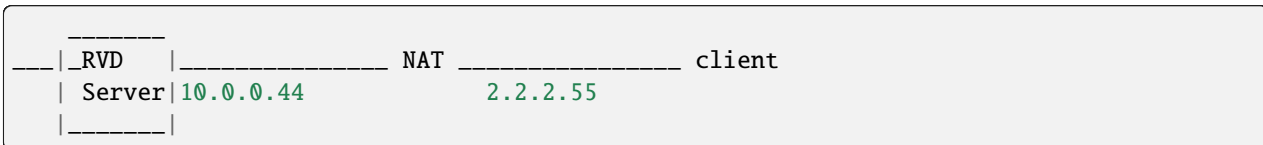
Since Ravada 0.10 you have to change it in the database:

```
mysql -u rvd_user -p ravada
mysql> select id,name,display_ip from vms;
+----+-----+-----+
| id | name       | display_ip |
+----+-----+-----+
| 1  | KVM_localhost |           |
| 2  | barrufet     |           |
+----+-----+-----+
mysql> UPDATE vms set display_ip='1.1.1.44' where id=1;
mysql> UPDATE vms set display_ip='1.1.1.55' where id=2;
```

### 3.57.2 NAT

The Ravada server can be behind a NAT environment. So the public address the client gets may not be the one configured in the Ravada server interface. You need to tell Ravada this IP. Then it will be shown to the end users for their client connection.

Example:



By default, the IP that is shown to the users will be 10.0.0.44 , but it is not accessible from outside.

For Ravada 0.9 and older set it at /etc/ravada.conf:

```
nat_ip: the external ip address
```

Since Ravada 0.10 you have to change it in the database:













You must configure the NAT IP in the Virtual Managers (VMs) table.

```
mysql -u rvd_user -p ravada
mysql> select id,name,nat_ip from vms;
+----+-----+-----+
| id | name       | nat_ip |
+----+-----+-----+
| 1  | KVM_localhost |       |
+----+-----+-----+
mysql> UPDATE vms set display_ip='2.2.2.55' where id=1;
```

If you have more nodes, each one probably will have its own public NAT IP. Set each one of them in the same way you set for the main node or Virtual Manager.

### 3.57.3 Auto Start

Virtual machines can be configured to start automatically when the physical host boots.

↓ Machine Name	<a href="#">hide clones</a>	Base	Public	Autostart	↑↓ Status	Actions
UbuntuBionic		✓	✓		Down	This Machine is a base
UbuntuBionic-abad [ Cloned ]		<input type="checkbox"/>		✓	Down	   
UbuntuBionic-jeff [ Cloned ]		<input type="checkbox"/>		<input type="checkbox"/>	Down	   
UbuntuBionic-troy [ Cloned ]		<input type="checkbox"/>		<input type="checkbox"/>	Down	   

You can enable the auto start column at the frontend configuration file at `/etc/rvd_front.conf`. Reboot the frontend with `systemctl restart rvd_front` to display the changes.

```
/etc/rvd_front.conf

{
  admin => {
    autostart => 1
  }
};
```

### 3.57.4 Choosing Storage Pool

#### Default Storage Pool

When creating virtual machines, Ravada chooses the storage pool with more free space available. If you want to force another, change it in the *Nodes setting* form in the frontend.

From *Admin Tools*, select *Nodes*. You will likely have only one : *KVM\_localhost*.

It may take a while to refresh this setting. Restart `rvd_back` running `systemctl restart rvd_back` if you want it applied right away.

#### Specific Storage Pools

Specific storages for bases and clones can be defined. This way you can use small and fast disk drives for bases and big but slower disks for clones.

**Warning:** If you set base and clone storages here, the default storage setting is ignored.

Add and define the storage pools as described in the [add kvm storage pool](#) manual. Then change the values in the node options screen.

**Warning:** These changes will apply to new virtual machines. Old virtual machines have already the disk volumes defined and can't be easily changed.

## node KVM\_localhost

options
bases
remove

name
KVM\_localhost
host
localhost
base storage
clone storage
default storage
default
cancel
save

### 3.57.5 Chek free memory ( from v0.3 )

Before start the domain, free memory of the Virtual Manager can be checked. This feature is only available in the development release.

First check the id field of the Virtual Manager in the table *vms*, then set the minimun of free available memory. In this example we require a minimun of 2 GB free:

```
mysql -u rvd_user -p ravada
mysql> select * from vms;
mysql> update vms set min_free_memory=20000000 where id=id*;
```

## 3.58 New documentation

We build documentation and host it in [Read the Docs](#).

All documentation files are stored only in *gh-pages* branch, with the following directory structure:

```
docs
├── _config.yml
├── devel-docs
├── docs
└── index.rst
```

Documentation is created using [reStructuredText](#) , is an easy-to-read, what-you-see-is-what-you-get plaintext markup syntax and parser system.

### 3.58.1 Procedure

1. Consider the editing style of existing pages.
2. Edit a doc page or create a new one in `gh-pages` branch.
3. Insert in `index.rst` according to the section.

---

**Note:** Documentation web is updated automatically, thanks to [Read the Docs](#).

---

### 3.58.2 Sidebar

The organization of the sidebar is configured in the `index.rst`. If you create a new documentation page remember to include in the section more according to the content. Add the `directory` and the name of `rst` file, e.g.:

`new_documentation.rst` in `docs/` will be `docs/new_documentation` somewhere in the `index.rst`

### 3.58.3 Convert POD files

Install `libpod-pom-view-restructured-perl` in your computer.

### 3.58.4 Step-by-step contributing to docs

So you want to contribute a documentation fix or new entry. Follow these 10 steps carefully.

1. Create a git account at [GitHub.com](#) if you don't already have one.
2. Go to the [Ravada github repository](#).
3. Create your own project copy clicking the `fork` button at the top right of the page
4. Configure your git account in your PC:

```
git config --global user.email "user@domain.com"
git config --global user.name "Your real name"
```

4. Download your copy `git clone https://github.com/YOUR_GITHUB_ACCOUNT/ravada`
5. Change to the github pages branch: `cd ravada ; git checkout gh-pages`
6. Edit the file inside the `docs/docs/` directory
7. If it is a new file run `git add new_file.rst`
8. Commit changes. It will ask for a one line description: `git commit -a`
9. Send the changes to github: `git push`
10. Go to [GitHub.com](#) and do a Pull Request. Make sure it is from the `gh-pages` branch

Branch: `gh-pages` ▼

New pull request

## 3.59 Configure LDAP Authentication

Ravada can use LDAP as the authentication engine.

### 3.59.1 Configuration

The configuration file is `/etc/ravada.conf`. The format is YML, make sure you edit this file with spaces, no tabs.

Add a section `ldap` like this:

```
ldap:
  server: 192.168.1.44
  port: 389 # or 636 for secure connections
  secure: 0 # defaults to 1 if port is 636
  base: dc=domain,dc=com
  admin_user:
    dn: cn=admin.user,dc=domain,dc=com
    password: secretpassword
```

The `_secure_` setting is optional. It defaults to 0 for port 389 (ldap) and to 1 for port 636 ( ldaps ). It can be enabled so secure connections can be forced for other ports.

The LDAP admin user can be a low level account with minimal privileges.

Another optional setting can be used to force the authentication method. By default Ravada tries first to bind to the LDAP as the user. If that fails then it tries to match the encrypted password. You can force the method with:

```
auth: all # defaults to all, can be all, bind, match
```

Notice matching authentication may be a security risk so bind is recommended.

### 3.59.2 Example: All users

All the users in the LDAP can have access to ravada:

```
ldap:
  server: 192.168.1.44
  port: 636
  base: dc=domain,dc=com
  admin_user:
    dn: cn=admin.user,dc=domain,dc=com
    password: secretpassword
```

### 3.59.3 Example: Group of users

Allow only a group of users to access ravada:

```
ldap:
  server: 192.168.1.44
  port: 636
  base: ou=users,ou=groupname,dc=upc,dc=edu
  admin_user:
    dn: cn=admin.user,dc=domain,dc=com
    password: secretpassword
```

### 3.59.4 Advanced Options

There are other options you can add to ldap:

- field: searches users using this field
- ravada\_posix\_group : allows only this group to access ravada
- sslversion : force the connection to LDAP with this SSL version: choose one of sslv2, sslv3, sslv23, tlsv1
- filter: filter attributes for the LDAP search

### 3.59.5 Example: Field

Add a new parameter field to search users with. In this example we search users with the cn attribute.

```
ldap:
  server: 192.168.1.44
  port: 636
  base: ou=users,ou=groupname,dc=upc,dc=edu
  field: cn
  admin_user:
    dn: cn=admin.user,dc=domain,dc=com
    password: secretpassword
```

### 3.59.6 Example: Posix Group

If you have all your users under a main OU (e.g. ou=users, dc=domain, dc=com), you can use Posix Groups (<https://ldapwiki.com/wiki/PosixGroup>) to create a list of users that can access to your Ravada instance, using their memberUid attribute. This allows you grant or remove access to ravada to some users without modify your LDAP structure.

```
ldap:
  server: 192.168.1.44
  port: 636
  base: ou=users,ou=groupname,dc=upc,dc=edu
  ravada_posix_group: cn=ravada,ou=groups,dc=domain,dc=com
  admin_user:
    dn: cn=admin.user,dc=domain,dc=com
    password: secretpassword
```



In the example, `cn=ravada,ou=groups,dc=domain,dc=com` is a Posix Group in your LDAP server. It should contain the `memberUid`'s of the users allowed to access to Ravada:

```
dn: cn=ravada,ou=groups,dc=domain,dc=com
objectclass: posixGroup
memberUid: user1
memberUid: user2
memberUid: user3
```

### 3.59.7 Example: Attribute Filter

In this example, only the users that have pass a filter can login:

```
ldap:
  server: 192.168.1.44
  port: 636
  base: dc=domain,dc=com
  filter: campus=North
  admin_user:
    dn: cn=admin.user,dc=domain,dc=com
    password: secretpassword
```

## 3.60 Authentication with Active Directory

Active Directory authentication can be configured using special LDAP attributes.

### 3.60.1 Configure Ravada

Add this entries to the file `/etc/ravada.conf`.

```
ldap:
  server: ad.server.ip
  port: 636
  base: cn=Users,DC=directory,DC=domain,DC=tld
  field: sAMAccountName
  admin_user:
    dn: cn=LDAP Query User,cn=Users,DC=directory,DC=domain,DC=tld
    password: xxxx
```

After changing the config file, restart the front service and try to login. *sudo systemctl restart rvd\_front*.

## Alternative login field

A typical user has both *full name* and *logon* fields. For example:

- Full Name ( `sAMAccountName` ) : Jhonny Xin
- Logon ( `userPrincipalName` ) : `joxin@mytestserver.com`

The basic configuration will allow the users login with their full name. You may prefer to use the logon field: *userPrincipalName*

```
ldap:
  server: ad.server.ip
  port: 636
  base: cn=Users,DC=directory,DC=domain,DC=tld
  field: userPrincipalName
  admin_user:
    dn: cn=LDAP Query User,cn=Users,DC=directory,DC=domain,DC=tld
    password: xxxx
```

Unfortunately, only one field can be used. We may add a new enhancement to use an alternate login field in a future release.

## Filter

With this optional setting only members of this group can login:

```
ldap:
  ...
  filter: memberOf=CN=ravada-users,CN=Users,DC=directory,DC=domain,DC=tld
```

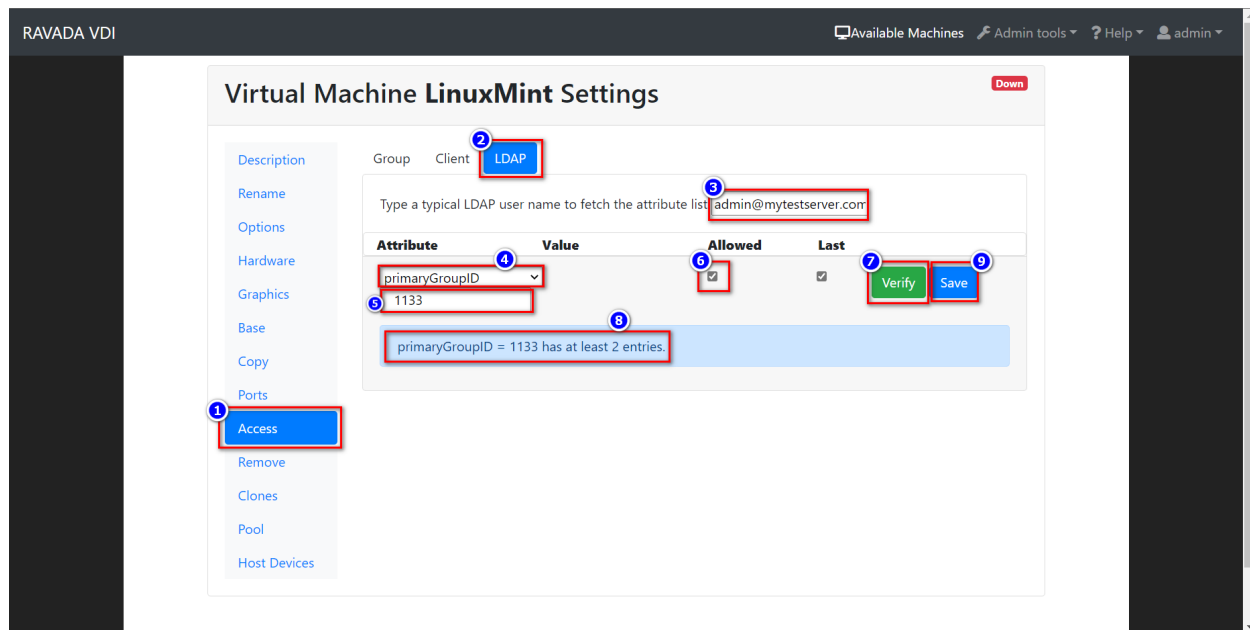
### 3.60.2 Group restriction

If you want that each user accessing VDI can be filtered according to group. In LDAP you may create groups and add the users to them. This is not possible with Active Directory, but you may filter the users with their *primary group id*.

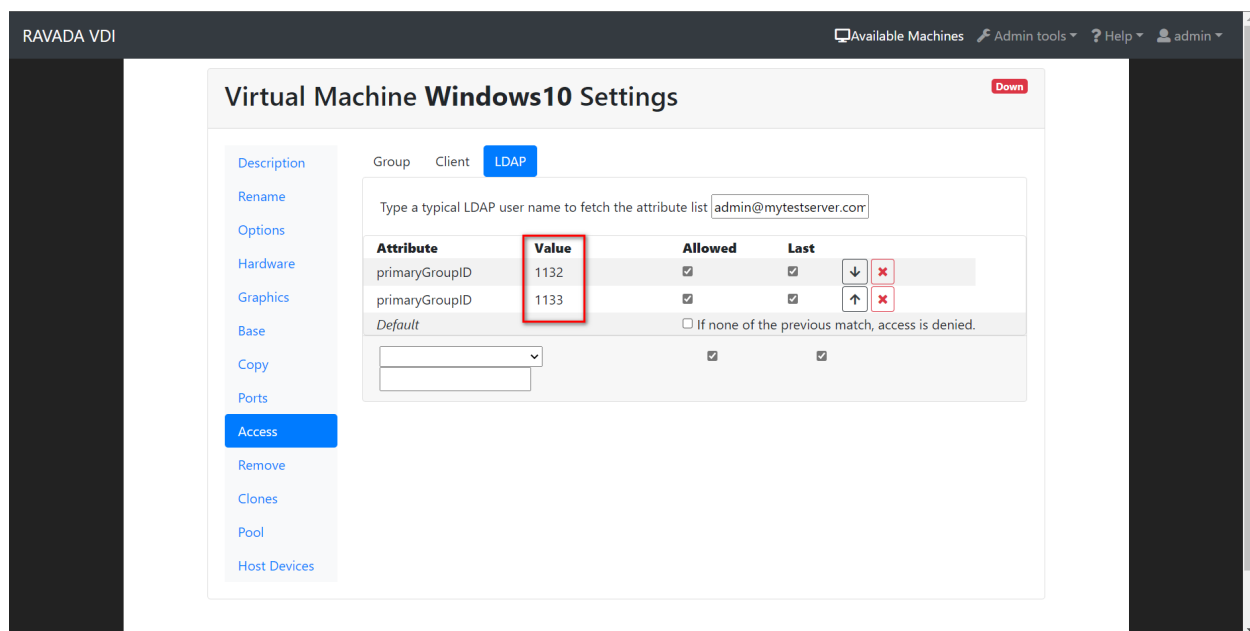
First of all you have to find out which is the primary group id for that user. You can inspect the LDAP attributes running in the console an LDAP test. This check tries to connect to the LDAP server, then it asks for an username and password. It tries to connect with that credentials and dumps the contents of this entry attributes. Search for *primaryGroupID*: `<number>` there. Also you could inspect in Windows AD using the attribute editor.

```
$ sudo rvd_back --test-ldap
```

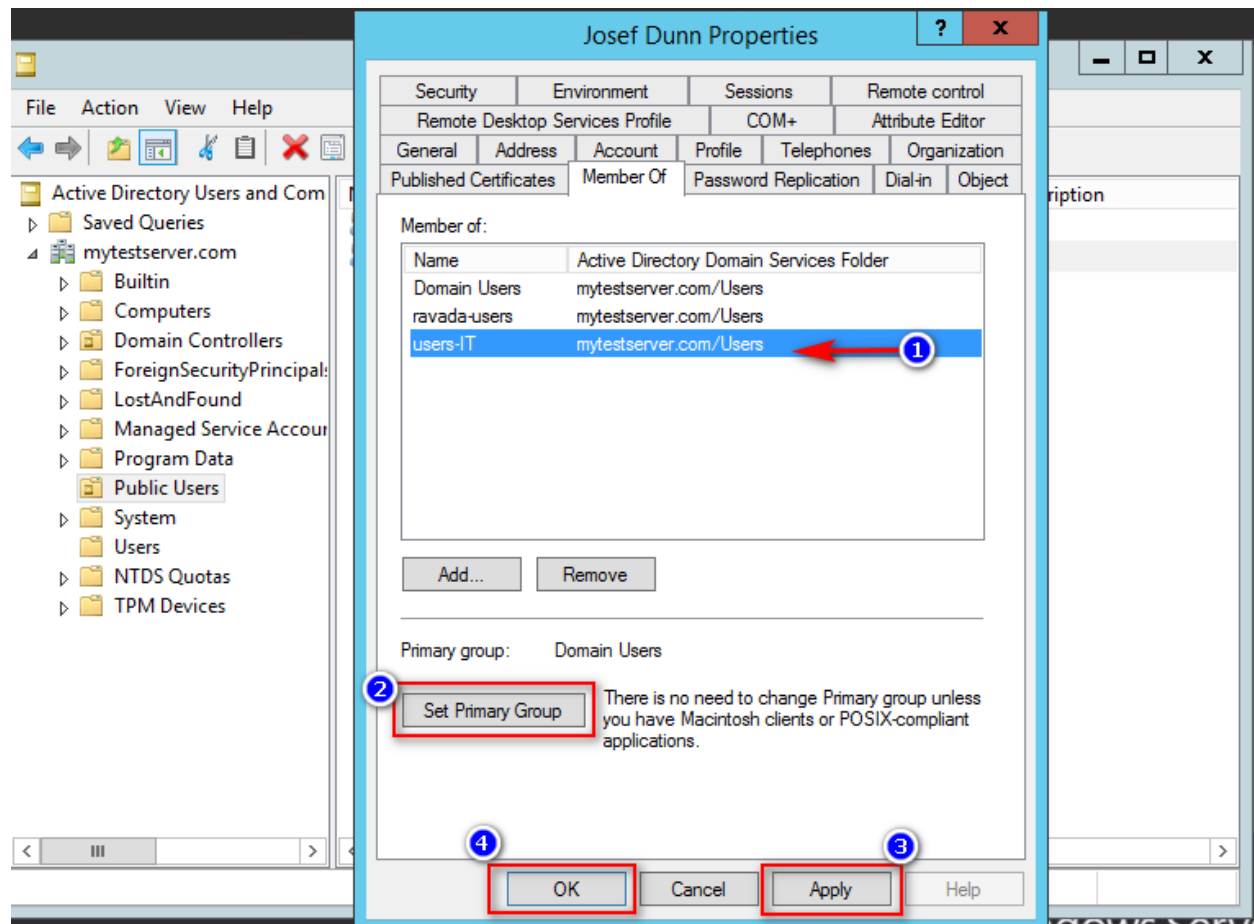
Now you have to configure the LDAP restrictions in the base virtual machine like this.



With this setup, we can limit every users Groups in Windows AD who want to access the VM.



Additionally, because *primaryGroupID* is a number showing the current users primary group. We should set it in Windows Server *Member Of* at the users setting menu:



Here users belong to 3 groups, but users-IT is their primary group ID, so when run `rvd_back -test-ldap` the result number in `primaryGroupID: <number>` will be users-IT ID group number. And this can be set as filter in Ravada Web.

### 3.60.3 Run

## 3.61 Set access restrictions to Virtual Machines

When a base is set as public, all the users have access to create clones by default. If you want to set access restrictions to base you can filter by LDAP attributes.

### 3.61.1 Access restrictions

Restrictions can be defined given a base and the value of an LDAP attribute. For example, you could set that only users who have the attribute "typology" as "teacher" are allowed to clone a virtual machine. Or you could deny "student" users access to another base.

### 3.61.2 Configuration

Actually there is a form to configure the access restrictions in the works. Until it is done you have to set this directly in the database.

#### Grant access

To grant access add an entry in the table `access_ldap_attribute` with the id of the base, the name of the attribute, the value of the attribute. The optional field `allowed` can be used to deny access to a virtual machine.

To remove a grant delete the row from the table `access_ldap_attribute`.

### 3.61.3 Examples

#### Example 1: grant access

Grant access to a virtual machine, only to those users that have `typology = teacher`.

First you need to know the id of the base virtual machine.

```
mysql> select id,name from domains where name = 'mymachine';
```

id	name
88	mymachine

Then add the restriction:

```
mysql> insert into access_ldap_attribute (id_domain, attribute,value) VALUES(88,'typology
↪','teacher');
```

#### Example 2: deny access

Deny access to a virtual machine, to those users that have `typology = student`.

First you need to know the id of the base virtual machine.

```
mysql> select id,name from domains where name = 'mymachine2';
```

id	name
89	mymachine2

Then add the restriction:

```
mysql> insert into access_ldap_attribute (id_domain, attribute,value,allowed) VALUES(89,
↪'typology','student',0);
```

### Example 3: remove an access restriction

We have some a restriction to deny access to students we want to remove because we want everybody access to that virtual machine:

```
mysql> select * from access_ldap_attribute;
+-----+-----+-----+-----+-----+
| id | id_domain | attribute | value | allowed |
+-----+-----+-----+-----+-----+
| 2 |      88 | typology | teacher |      1 |
| 3 |      89 | typology | student |      0 |
+-----+-----+-----+-----+-----+
mysql> delete from access_ldap_attribute where id=3;
```

## 3.62 Test Active Directory

This document is a guide to test Active Directory features from Ravada source.

If you want to test the active directory you must create 2 files: one for the ravada configuration parameters, and another one with AD user and password to test.

### 3.62.1 Ravada Conf

Create a file : t/etc/ravada\_ad.conf with these contents:

```
ActiveDirectory:
  host: thehost
  port: 389
  domain: thedomain
  principal: theprincipalwhatever it is
```

### 3.62.2 Auth data

It is required a valid user and password to test AD. Put them in the file t/etc/test\_ad\_data.conf

```
name: theusername
password: thepassword
```

### 3.62.3 Run the tests

From the source root directory run:

```
$ perl Makefile.PL
$ make && prove -b t/67_user_ad.t
```

## 3.63 Tuning Ravada

Some additional settings to improve your Ravada installation.

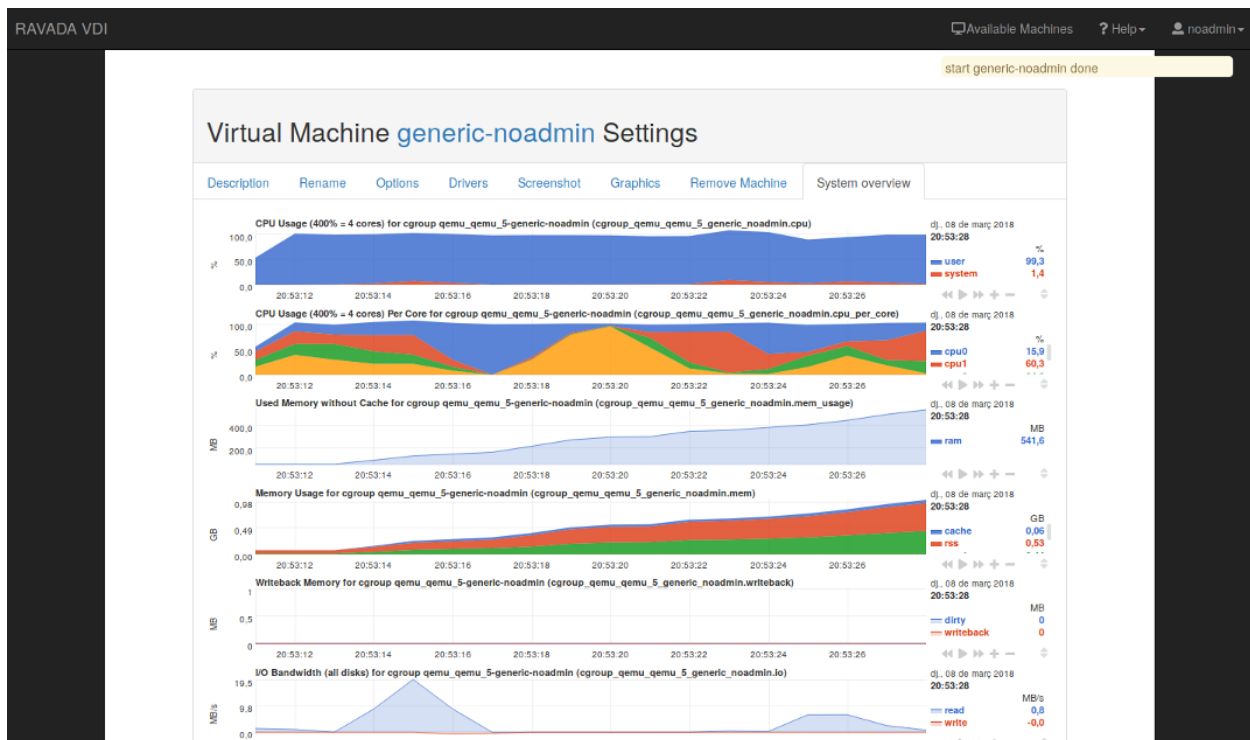
### 3.63.1 IP Conflict

When there are machines that have been hibernated long ago, their ip address may conflict with other that have just been created. That is because the IP addresses have a short life span by default.

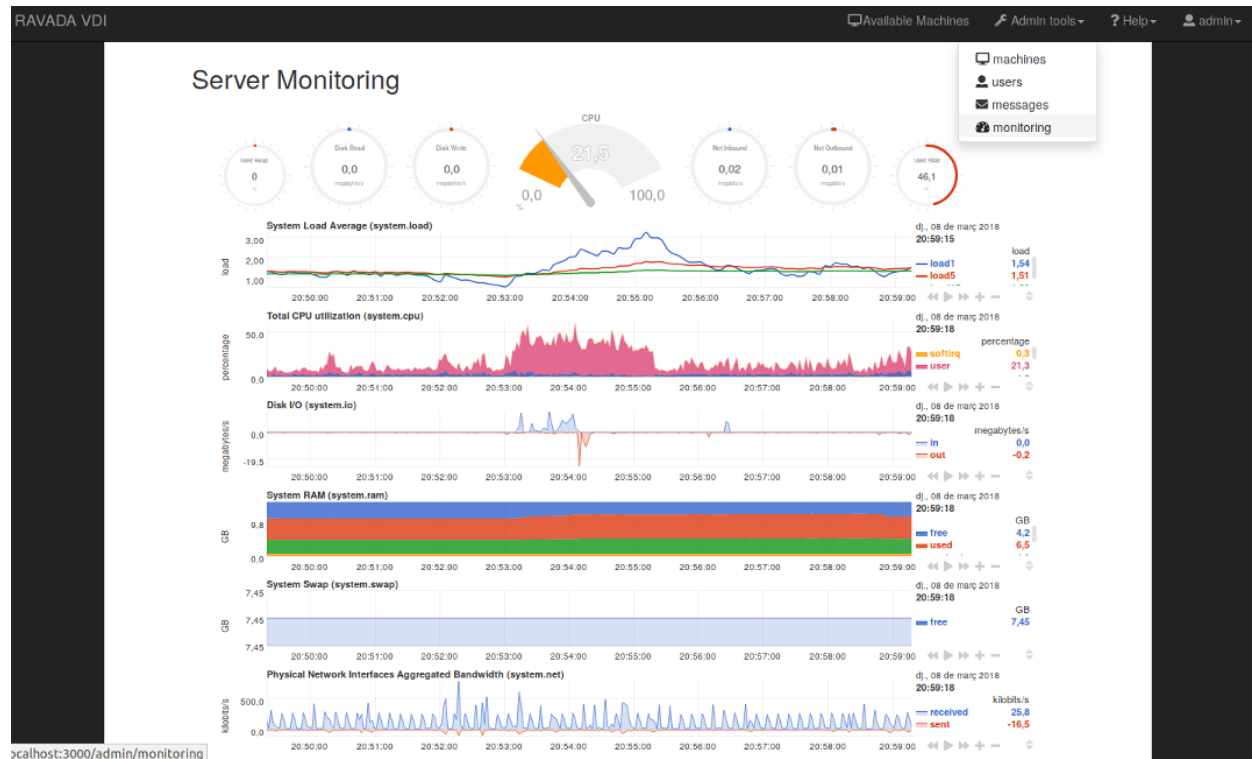
**Warning:** TODO Make IP addresses last forever changing the settings of dnsmasq.

## 3.64 Server Monitoring

From VM settings tab you can see the VM system overview:



And for admin role you can see the server:



Server monitoring is disabled by default. In order to support Ravada server monitoring, you have to do a few steps:

- install [netdata](#) on a ravada server
- and enable monitoring in `/etc/rvd_front.conf`

### 3.64.1 Install my-netdata.io

Follow this steps from [my-netdata.io](#)

or execute this on a terminal:

```
# bash <(curl -Ss https://my-netdata.io/kickstart-static64.sh)
```

Linux 64bit, pre-built static binary installation for any Linux distro, any kernel version - for Intel/AMD 64bit hosts.

### 3.64.2 Apache config for netdata with SSL

Enable SSL and proxy in apache:

```
# a2enmod proxy_http proxy ssl
# a2ensite default-ssl
```

In `/opt/netdata/etc/netdata/netdata.conf` add:

```
[web]
default port = 19998
```

In `/etc/apache2/ports.conf` add:



```
<IfModule ssl_module>
    Listen 443
    Listen 19999
</IfModule>
```

and adding a new virtualhost for port 19999 in `/etc/apache2/sites-available/default-ssl.conf`

```
<VirtualHost *:19999>
    ProxyRequests Off
    ProxyPreserveHost On
    ProxyPass / http://localhost:19998/ keepalive=On
    ProxyPassReverse / http://localhost:19998/

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    SSLEngine on
    SSLCertificateFile      /etc/ssl/certs/server.crt
    SSLCertificateKeyFile   /etc/ssl/private/server.key
    SSLCertificateChainFile /etc/ssl/certs/ca.crt
</VirtualHost>
```

**Warning:** Be careful with self-signed certificates. The browser needs to accept the certificate. We recommend the use of [Let's Encrypt](#) or your trusted SSL provider.

Then restart netdata and apache:

```
# systemctl restart apache2
# systemctl restart netdata
```

Thanks to [@jlopezramos](#) for this contribution.

### 3.64.3 Enable monitoring

Monitoring is disabled by default. Add in `/etc/rvd_front.conf` file:

```
monitoring => 1
```

Restart rvd\_front service:

```
# systemctl restart rvd_front.service
```

### 3.64.4 Tuning netdata

#### Disable mail alarms

Edit the file `/opt/netdata/etc/netdata/health_alarm_notify.conf`, `/usr/lib/netdata/conf.d/health_alarm_notify.conf` or `/etc/netdata/health_alarm_notify.conf` and set

```
SEND_MAIL="NO"
```

### 3.64.5 Monitoring Architecture

All servers have a local [My-netdata](#) installation, this lets us know what happens now. Metrics in real time, small story. The file of metrics is stored in the [Graphite](#) server, this allows us to obtain historical and archived data. And [Grafana](#) lets us know what happened in the past. Low detail metric, long history. Follow this [link](#) to know more about it.

### 3.64.6 Graphite backend

Edit the file `/opt/netdata/etc/netdata/netdata.conf`:

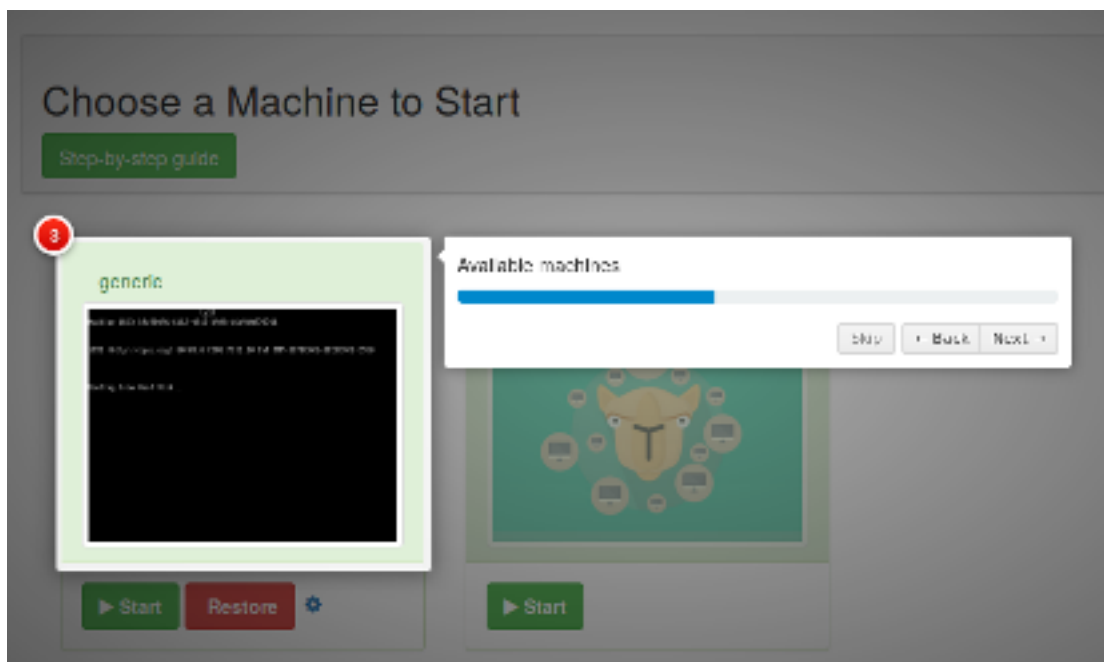
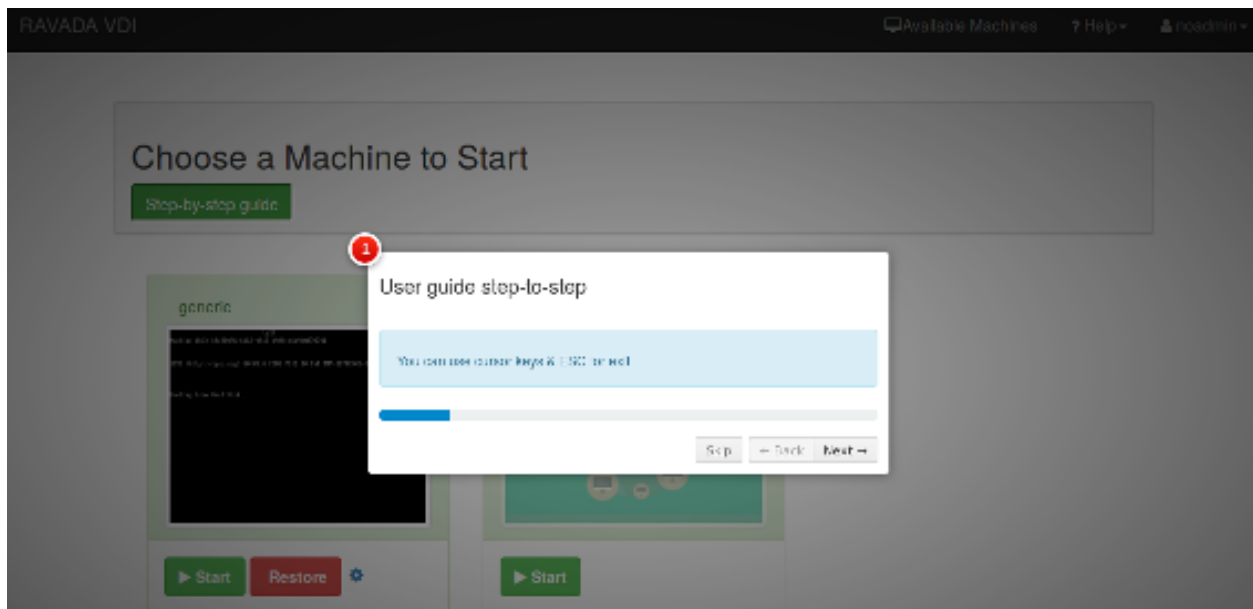
```
[backend]
    host tags =
    enabled = yes
    data source = average
    type = graphite
    destination = <GraphiteServer>
    prefix = netdata
    hostname = <hostname>
    update every = 10
    buffer on failures = 10
    timeout ms = 20000
    send names instead of ids = yes
    send charts matching = *
    send hosts matching = localhost *
```

### 3.64.7 Graphana

This [dashboard](#) makes our lives easier.

## 3.65 Guide button step-by-step

The guide button offers step-by-step help. You can see a taste in the following screenshots:



Guide button is disabled for default. In order to support you have to enable guide in `/etc/rvd_front.conf`

### 3.65.1 Enable guide

In `/etc/rvd_front.conf`:

```
guide => 1
```

Restart `rvd_front` service:

```
systemctl restart rvd_front.service
```

### 3.65.2 Custom guide

Add path in `/etc/rvd_front.conf`:

```
guide_custom => '/js/custom/custom_intro.js'
```

## 3.66 Offline libraries in frontend

Javascript and CSS libraries can be configured in the local server instead of using a CDN.

- CDNs deliver faster loading speeds for users
- CDNs lower the cost of bandwidth.
- Improved SEO
- Handling spikes in traffic

But some users behind firewalls may experience rendering problems, from China i.e. By default we use CDN versions of the JavaScript and CSS libraries. In some specific cases it may be useful to have those libraries in the same server as the Ravada web frontend runs.

Since Ravada 0.5 release we package the required javascript and CSS files. You can enable the local copy setting the file `/etc/rvd_front.conf` in your host.

```
fallback => 1
```

and restart the `rvd_front.service` to apply changes.

```
# systemctl restart rvd_front.service
```

Refresh your browser cache and now Ravada use JS and CSS downloaded from your own server.

## 3.67 Exposing a Virtual Machine

By default the virtual machines are created inside a private internal network. This allows the user to reach internet but no connections from outside are allowed.

Some times we may want to install a server in the Virtual Machine and grant access to it. There are many ways to expose a Virtual Machine and allow access to it from outside. Here we describe a few: set a public IP address, redirect with IPTables and HTTP forwarding.

### 3.67.1 Setting a public IP

One way to expose the virtual machine is use a public IP instead the private used by default. To do so you have to manually edit the machine definition before creating the base. Change the network settings to [bridge](#).

This setting gives the more exposure to the virtual machine, so firewalls and other security measures must be configured.

### 3.67.2 IPTables redirection

You can redirect a port from the host or a virtual machine acting as gateway to the internal address of the machine you want to expose.

This technique restricts which ports from the internal machine are exposed from outside.

### 3.67.3 HTTP Forwarding

HTTP forwarding can be configured in a web server in the host to access internal web services from outside.

### 3.67.4 Expose Ports

Ports from the virtual machine can be exposed to outside with this new feature introduced in release 0.5.

## 3.68 Exposing Ports from a Virtual Machine

Since release 0.5 ports from the internal virtual machine can be exposed to outside.

### 3.68.1 Requirements

Ports exposing works with Linux Kernel *iptables* so it requires the host to have it installed in properly configured.

The virtual machine must have a way to tell the host what is the internal IP to the host, so the *qemu guest agent* must be installed.

### 3.68.2 Configure Expose

From the virtual machine settings there is a new menu item **Ports**. There you can add new ports to be exposed.

There you must add the number of the internal port to be exposed and an optional name. If you set this port to *restricted* it will only be allowed to the remote client IP.

#### Fields

- *Port*: internal port in the virtual machine
- *Name*: optional short description of the service to expose
- *Restricted*: restrict the port to the remote client if set

### Virtual Machine **ubuntu** Settings

Description  
Rename  
Options  
Drivers  
Hardware  
Graphics  
Base  
Copy  
**Ports**  
Access  
Remove  
Clones  
Pool

Action	Port	Restricted	Name	Public Port
	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>	
✕	22	<input checked="" type="checkbox"/>	ssh	5995
✕	80	<input type="checkbox"/>	http	5996

### 3.68.3 Run

When the virtual machine gets started Ravada searches for a free port in the host to expose the port through.

In this example the user can connect to the IP 10.1.36.68 port 5954 to access the SSH server inside the virtual machine.

## Running ubuntu-frankie

A viewer is required to run the virtual machines. [Read more.](#)

### Machine Information

- Internal IP: 192.168.122.227
- Max Memory: 1047552
- Memory: 1047552
- CPUs: 1

### Open ports

- ssh 10.1.36.68:5954 → 22
- http 10.1.36.68:5955 → 80

[view](#)

*Press SHIFT + F12 to exit the virtual machine*

Ports redirection requires the internal IP of the virtual machine to be active. So it retries for a while trying to redirect until an IP is detected or it fails.

---

**Note:** If your exposing ports don't work. Check if the port is up and if you have the [Qemu Guest Agent](#) installed on the guest machine.

---

### 3.68.4 Settings

Internal ports are shown in the host from a starting port number. It can be changed from the global settings page in the *Port Expose* section.

### 3.68.5 TroubleShooting

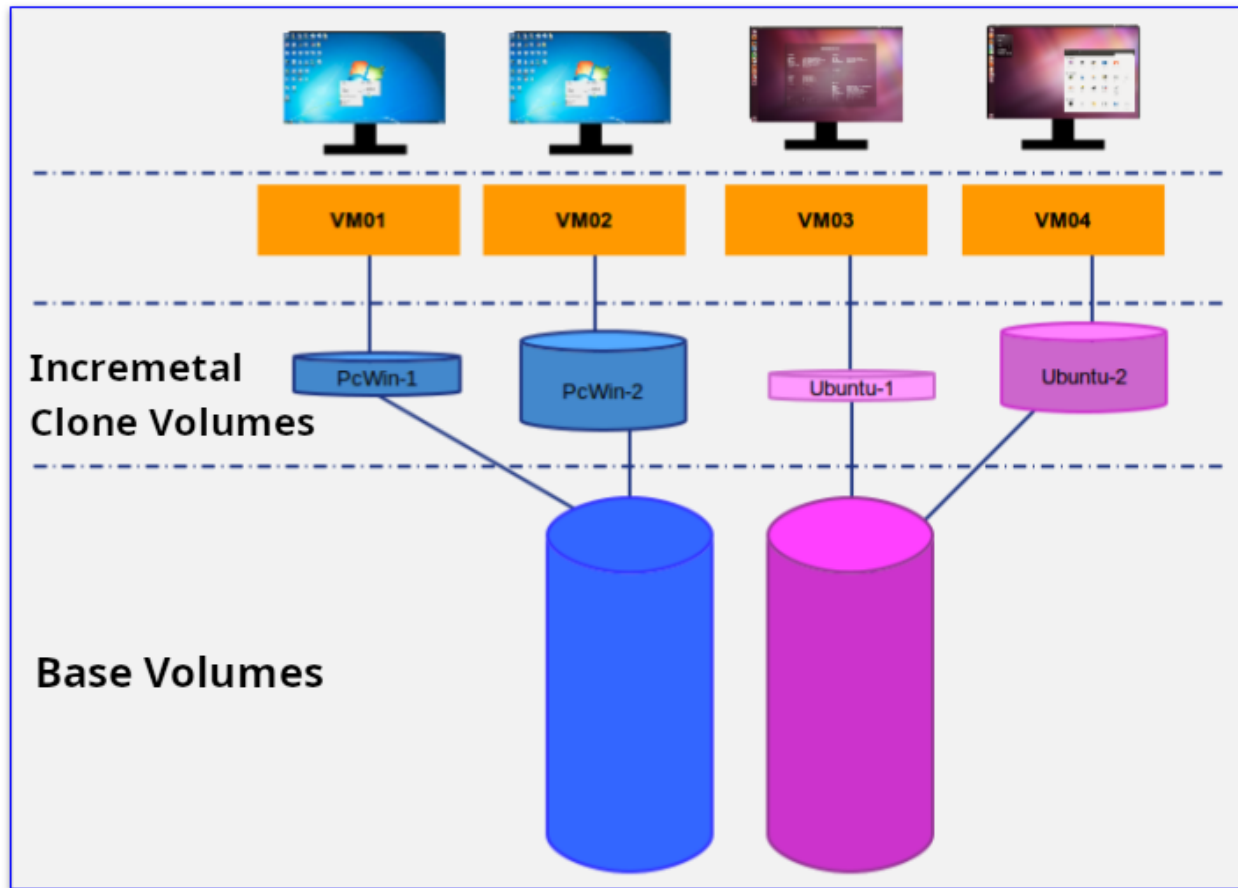
*The port appears down when I start the virtual Machine*

Solution: This port has a service in the virtual machine that is not up. Be sure the service has started, and that it is listening in the correct port. Check also if this virtual machine has an internal firewall that may block connections.

## 3.69 Keeping the Base updated

### 3.69.1 Base volumes

Once a base is prepared and clones are made the base becomes *read only*. This is required because the clones disk volumes depend of the data stored in the bases. So it is not possible to change the base because the clones storage would become corrupted.



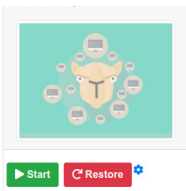
### 3.69.2 Rebasing

There is a way to keep the base updated for all the clones. This requires you create a new base from the old one and make all the clones depend from the new one. This is called *rebase* and requires careful preparation in advance.

### 3.69.3 Virtual Machine Volumes

#### Restoring clones

Before showing the virtual machine volumes, we must explain the restore procedure.



When an user requests a *restore* of a clone. This virtual machine system information get removed. When it starts again the original content created in the base will show up.

This is different than removing the Virtual Machine. If it is removed all the content and the Virtual Machine itself will be deleted. So just restored machines still are shown in the lists.



## Volume Types

There are three kinds of disk volumes that can be assigned to the Virtual Machines:

- System: This is the normal storage. Removed on restore.
- Temporary: Content will be cleaned on restore and shutdown
- Data: Content will be kept on restore

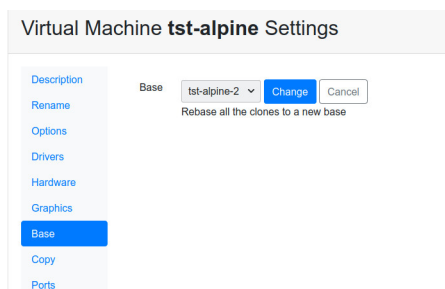
So you should use temporary volumes for *swap*, *pagefile* or other temporary files. Use data volumes to store users files and other information that doesn't depend of the operative system.

When you *rebase* a virtual machine, system volumes will be replaced and *data volumes will be kept untouched*.

### 3.69.4 Rebase requirements

When you rebase, the new base virtual machine must have the same storage outline as the old base. A good practice is to clone the old base, do some changes and make it the new base.

After the rebase procedure is completed, the operative system will be replaced, the temporary files would be restored to their initial state and the data volumes will be exactly like they were before rebasing.












Run rebase on the virtual machine options at the base screen. You can rebase only one clone or request all the clones to be updated. This will take a few minutes. It depends of the size of the virtual machine volumes.

## 3.70 Spinoff Clone

Spinoff clone releases a virtual machine from its bases. It becomes independent and you can use it on its own.

### 3.70.1 Clones

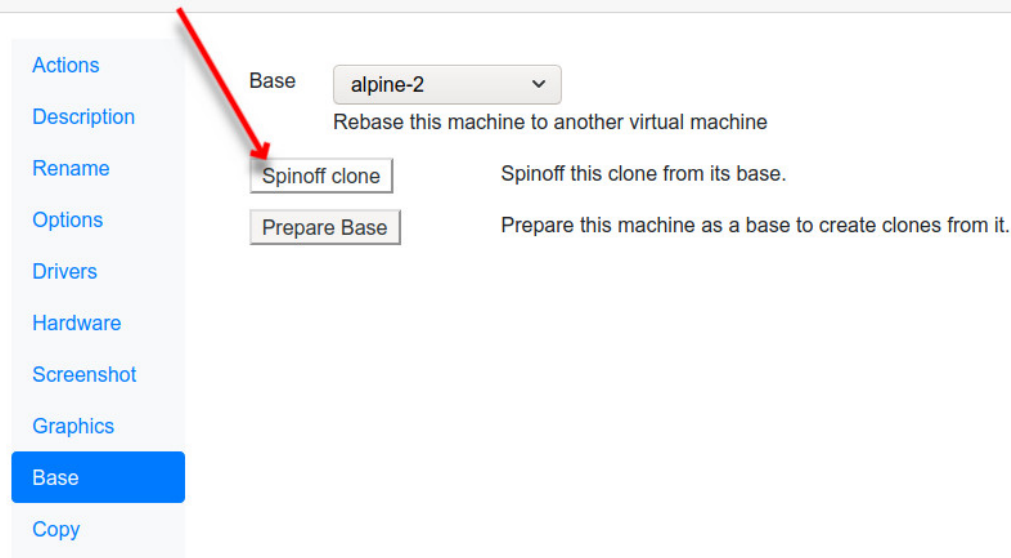
In this example we have bases with nested bases and clones.

alpine 	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 alpine-1	<input type="checkbox"/>	
 alpine-2 	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 alpine-2-1	<input type="checkbox"/>	
 alpine-2-2	<input type="checkbox"/>	
 alpine-2-3	<input type="checkbox"/>	
 alpine-2-4	<input type="checkbox"/>	
 alpine-3	<input type="checkbox"/>	
 alpine-4	<input type="checkbox"/>	
 alpine-5	<input type="checkbox"/>	

### 3.70.2 Proceed to Spinoff

We want to spinoff the virtual machine *alpine-2-1*, so we click in its name and then in the *Base*” tab with press *\*Spinoff clone*.

## Virtual Machine **alpine-2-1** Settings



Base **alpine-2** ▾  
Rebase this machine to another virtual machine

**Spinoff clone** Spinoff this clone from its base.

**Prepare Base** Prepare this machine as a base to create clones from it.

Actions  
Description  
Rename  
Options  
Drivers  
Hardware  
Screenshot  
Graphics  
**Base**  
Copy

After confirm this is what we want Ravada will proceed to spinoff the clone from all its bases.

## Virtual Machines

Requests **2**

Machine Name	Base
<b>aa-1</b>	<input checked="" type="checkbox"/>
<b>alpine</b> +	<input checked="" type="checkbox"/>
<b>alpine-2-1</b>	<input type="checkbox"/>
<b>focal</b> +	<input checked="" type="checkbox"/>

### 3.70.3 What happened behind the scenes ?

These are the disk volumes for the machine *alpine-2-1* before the spinoff. All of them are base-chained to *alpine-2* and then to *alpine*.

Volume files before spinoff:

```
<source file='/home/images.2/alpine-2-1-vda.alpine-2-av-vda.qcow2' />
  <source file='/home/images.2/alpine-2-av-vda.ro.qcow2' />
    <source file='/home/images.2/alpine-vda.ro.qcow2' />
```

Volume files after spinoff:

```
<source file='/home/images.2/alpine-2-1-vda.alpine-2-av-vda.qcow2' index='4' />
```

When we inspect the disk files volumes after the spinoff we can see it has no dependencies. So its contents are not incremental from other disk volumes.

### 3.71 Live web Chatwoot

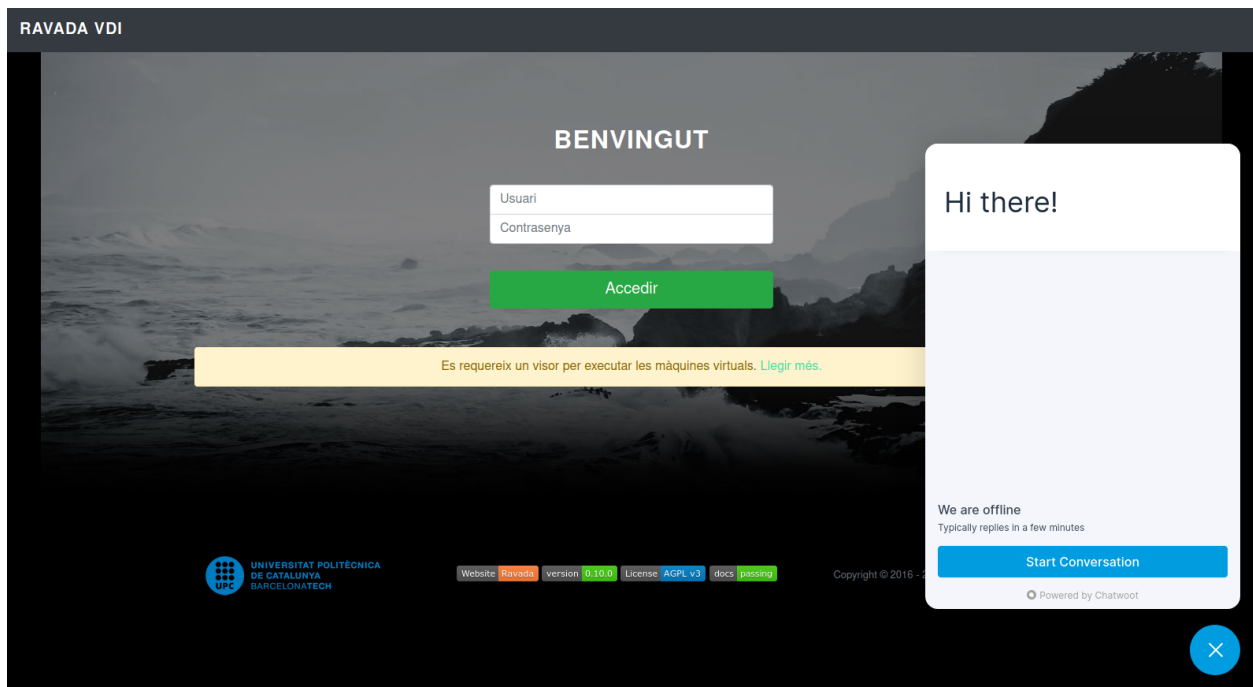
If you want to offer a communication channel for users. We propose this simple and powerful solution.

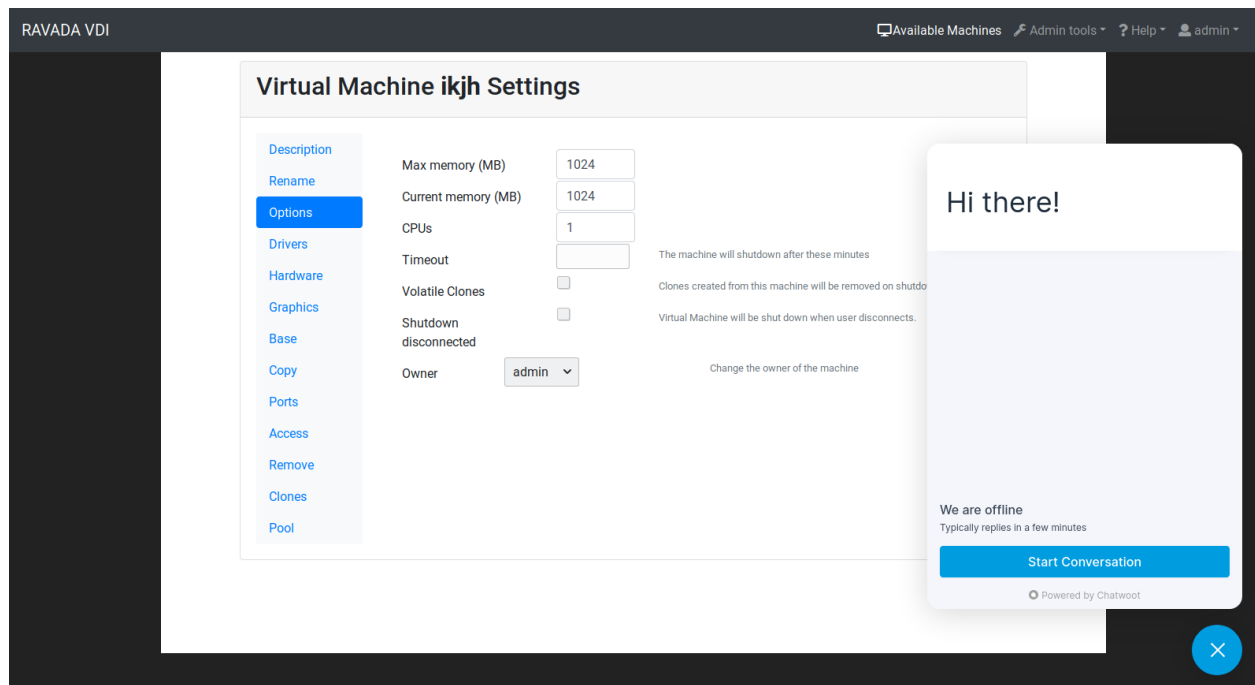
**Chatwoot** is live chat software. It's open source and has a great community. You can access the code [here](#).

You need a Chatwoot server, you have different options. If you are interested in self-hosted follow this [guide](#).

Here you will not find a [chatwoot manual](#), only a few steps to embed your widget code.

The Widget appears at the login script, main page and startup screens:





Once you have the widget you have to paste it in a *js* file, and add a config to Ravada.

### 3.71.1 Create a JS widget

Copy your code in the file: `/usr/share/ravada/public/js/custom/chatwoot_widget.js`. Do not copy the `<script>` tags, only the javascript code. Set your chatwoot server at the second line.

```

1  (function(d,t) {
2      var BASE_URL = "https://chatwoot.server.example.org";
3      var g=d.createElement(t),s=d.getElementsByTagName(t)[0];
4      g.src= BASE_URL + "/packs/js/sdk.js";
5      s.parentNode.insertBefore(g,s);
6      g.onload=function(){
7          window.chatwootSettings = {
8              locale: 'ca',
9              type: 'expanded_bubble',
10             launcherTitle: 'Some message',
11             showPopoutButton: true
12         }
13         window.chatwootSDK.run({
14             websiteToken: 'xxxxxxxx4Yh7RkXPtt1',
15             baseUrl: BASE_URL
16         })
17     }
18 })(document,"script");


```

### 3.71.2 Configuration

Add the widget path in the frontend settings. Go to *Admin Tools* , *Settings* You also need to allow third party content inside HTML, set the server address in the *Content Security Policy*.

## Global Settings

### Frontend


Maintenance	<input type="checkbox"/> Admin users can still log in from <a href="/login">/login</a>
Widget 	<input type="text" value="/js/custom/chatwoot_widget.js"/>
Content Security Policy	<input type="text" value="chat.server.example.org"/>
	<input type="checkbox"/> Advanced CSP

### Content Security Policy

If you want to fine-tune the Content Security Policy, check the *Advanced CSP* to display more options.

## Global Settings

### Frontend

Maintenance	<input type="checkbox"/> Admin users can still log in from <a href="/login">/login</a>
Widget 	<input type="text" value="/js/custom/chatwoot_widget.js"/>
Content Security Policy	<input checked="" type="checkbox"/> Advanced CSP

all	<input type="text" value="chat.server.example.org"/>
connect-src	<input type="text"/>
default-src	<input type="text"/>
font-src	<input type="text" value="font.example.org"/>
frame-src	<input type="text"/>
media-src	<input type="text"/>
object-src	<input type="text"/>
script-src	<input type="text"/>
style-src	<input type="text"/>

## Apply

There is no need to restart the server. It is applied right after saving.

## Check headers

Use curl to check the content security policy headers

```
$ curl --head https://server.example.org
```

## 3.72 PCI GPU Device Passthrough

In this document we show how to add a NVIDIA GPU to a KVM virtual Machine in Ravada VDI. Any PCI device could be added in a similar way.

---

**Note:** We will not use the GPU as a display device in this guide. We will only try to run GPU calculations with CUDA.

---

### 3.72.1 Status

Please note that this is a very active topic and the instructions outlined here might not work in your environment

### 3.72.2 Requirements

- One or more NVidia GPUs
- A properly configured kernel along with a recent version of qemu. We were successful with Ubuntu 20.04 with kernel 5.8.
- Ravada version 2.0. It is currently labelled as *alpha* but it is being used in production in our own servers.

### 3.72.3 Hardware Identification

```
$ sudo lspci -Dnn | grep NVIDIA
$ 0000:1b:00.0 VGA compatible controller [0300]: NVIDIA Corporation Device [10de:2204] (rev a1)
$ 0000:1b:00.1 Audio device [0403]: NVIDIA Corporation Device [10de:1aef] (rev a1)
```

As you can see there are two devices attached to this PCI, that is because the video card has also an audio device. From here we gather:

- PCI data: 0000:1b:00.0 and 0000:1b:00.1
- Devices: 10de:2204 and 10de:1aef

### 3.72.4 OS Configuration

This is the configuration for the host server. We must setup the kernel so the PCI device is not used in any way so it can be passed to the virtual machine.

#### Grub

Add this to GRUB\_CMDLINE\_LINUX line in */etc/default/grub*. This is suited for Intel computers. Notice we added the devices identified in the previous step to *pci-stub*.

Here we are preventing the GPU to be used by the main host so it can be passed to the virtual machine.

```
GRUB_CMDLINE_LINUX_DEFAULT="snd_hda_intel.blacklist=1 nouveau.blacklist=1 intel_iommu=on,
↪rd.driver.pre=vfio-pci pci-stub.ids=10de:2204,10de:1aef"
```

AMD require a similar setup adding

```
GRUB_CMDLINE_LINUX_DEFAULT="snd_hda_intel.blacklist=1 nouveau.blacklist=1 amd_iommu=on,
↪iommu=pt kvm_amd.npt=1 kvm_amd.avic=1 rd.driver.pre=vfio-pci pci-stub.ids=10de:2204,
↪10de:1aef"
```

#### Modules

Blacklist modules creating the file */etc/modprobe.d/blacklist-gpu.conf*

```
blacklist nouveau
blacklist radeon
blacklist amdgpu
blacklist snd_hda_intel
blacklist nvidiafb
```

#### GPUs with embedded USB

If your *GPU* also sports an *USB device*, you need to check what driver this device is using and blacklist it here also.

**Warning:** This procedure will render all the USB devices of the server unavailable. From now on the USB keyboard can not be used anymore. Make sure you always have access with ssh. In case you needed to use the keyboard again you have to remove the *blacklist xhci\_pci* from the file */etc/modprobe.d/blacklist-gpu.conf*, issue *update-grub* and reboot the server.

Use *lspci -Dnn -k* to check what driver it is using.

```
$ lspci -Dnn -k
000:b1:00.2 USB controller [0c03]: NVIDIA Corporation TU104 USB 3.1 Host Controller,
↪[10de:1ad8] (rev a1)
Subsystem: NVIDIA Corporation TU104 USB 3.1 Host Controller [10de:129f]
Kernel drivers in use: xhci_pci
```

Then add it to the end of the file */etc/modprobe.d/blacklist-gpu.conf*:



```
blacklist xhci_pci
```

### Make drivers use vfio

Also prevent the nvidia drivers to load and raise vfio-pci instead in /etc/modprobe.d/nvidia.conf

```
softdep nouveau pre: vfio-pci
softdep nvidia pre: vfio-pci
softdep nvidia* pre: vfio-pci
```

Add the modules so they are loaded on boot in /etc/modules

```
vfio
vfio_iommu_type1 allow_unsafe_interrupts=1
vfio_pci ids=10de:2204,10de:1aef
vfio_virqfd
```

Pass the device identifiers to vfio-pci in /etc/modprobe.d/vfio.conf. This is duplicated from the previous step, it may be removed eventually from this doc if we find it unnecessary. Anyway it doesn't harm.

```
options vfio-pci ids=10de:2204,10de:1aef disable_vga=1
```

When loading KVM make it ignore MSRS in /etc/modprobe.d/kvm.conf options kvm ignore\_msrs=1

```
options kvm ignore_msrs=1
```

Add id of the device used by nvidia at /etc/initramfs-tools/modules. This must be one line:

```
vfio vfio_iommu_type1 vfio_virqfd vfio_pci ids=10de:2204,10de:1aef,10de:1ad8 allow_
↳ unsafe_interrupts=1
```

Update the grub and initram configuration and reboot.

```
$ sudo update-grub
$ sudo update-initramfs -u
$ sudo reboot
```

## 3.72.5 Checks

### Modules

No nvidia nor nouveau should be loaded:

```
$ sudo lsmod | egrep -i "(nouveau|nvidia)"
```

The device should use vfio driver:

```
$ lspci -k | egrep -A 5 -i nvidia
```

```
1b:00.0 VGA compatible controller: NVIDIA Corporation Device 2204 (rev a1)
  Subsystem: Gigabyte Technology Co., Ltd Device 403b
  Kernel driver in use: vfio-pci
  Kernel modules: nvidiafb, nouveau
1b:00.1 Audio device: NVIDIA Corporation Device 1aef (rev a1)
```

(continues on next page)

(continued from previous page)

```
Subsystem: Gigabyte Technology Co., Ltd Device 403b
Kernel modules: snd_hda_intel
```

See that though in the NVIDIA VGA the preferred kernel modules are nvidiafb and nouveau, it actually loads vfio-pci which is great.

## IOMMU

Check it is enabled

```
$ dmesg | grep -i iommu | grep -i enabled
```

```
[ 0.873154] DMAR: IOMMU enabled
```

Verify the iommu groups. Both devices should be in the same group. We use *grep* to search for the PCI device numbers we found in the very first step.

```
$ dmesg | grep iommu | grep 1b:00
```

```
[ 2.474726] pci 0000:1b:00.0: Adding to iommu group 38
[ 2.474807] pci 0000:1b:00.1: Adding to iommu group 38
```

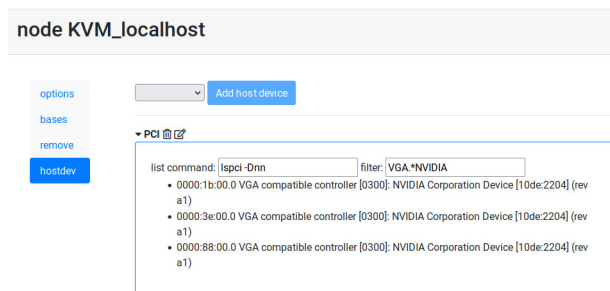
### 3.72.6 Ravada Setup

Now we want to use the GPU, by now we will only try to execute CUDA so it will not be a device used for display. This can also be achieved but it will be addressed in future releases.

After we have the host configured we must tell Ravada we want to pass some PCI devices to the virtual machines.

#### Configure the Node Host Device

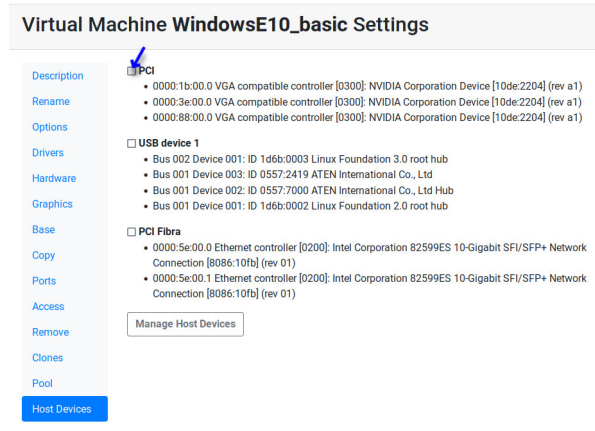
At the node configuration we add a PCI Host Device group. This is a pool of devices that will be added to the clones. In this example we select PCI and then click on “Add host device”.



After a few seconds we can see the PCI devices available in the host, we filter only the Nvidia brand.

Now the Host Device will be available in the Hardware configuration in the virtual machine.

Now, when the virtual machine is started it will pick one of the free devices and it will appear as a PCI entry.



### 3.72.7 Virtual Machine GPU Ubuntu setup

As an example we load the GPU in Ubuntu and verify it is being used.

#### Packages

Configure from the graphical interface to load proprietary drivers for NVIDIA server.

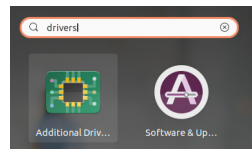


Fig. 3: Search for the additional drivers application  
Press the Windows key and type additional , click in the application called *Additional Drivers*

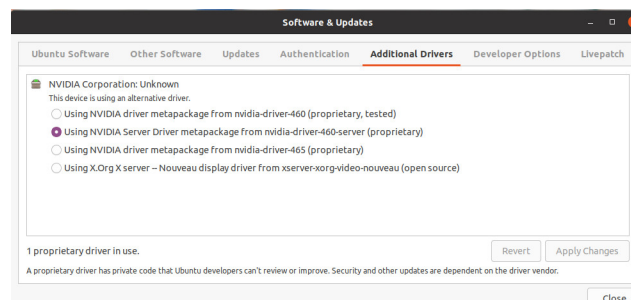


Fig. 4: Choose the NVIDIA driver for servers  
In our scenario we only want to run CUDA on the GPU so we just select the server drivers.

This is the list of packages for our setup:

- nvidia-compute-utils-460-server
- nvidia-dkms-460-server
- nvidia-driver-460-server
- nvidia-kernel-common-460-server

- nvidia-kernel-source-460-server
- nvidia-settings
- nvidia-utils-460-server

### Choose the Display VGA

After installing the NVidia drivers the Window Manager may try to run on top of the GPU and fail. Choose the other video card:

First let's what cards do you have:

```
$ sudo prime-select
Usage: /usr/bin/prime-select nvidia|intel|on-demand|query
```

Choose not nvidia, in our case it is intel:

```
$ sudo prime-select intel
```

Add the nvidia module to load on startup. Check there is this line in /etc/modules

```
nvidia_uvm
```

Reboot the virtual machine now. The other VGA should be used for display and the NVIDIA GPU can be used to run other tasks.

### 3.72.8 CUDA

In this particular installation we wanted to try CUDA. We install the package and check if it works:

```
$ sudo apt install nvidia-cuda-toolkit
$ nvidia-smi
```

If it works nvidia smi will show the detected hardware:

```
Driver Version           : 460.73.01
CUDA Version             : 11.2
Attached GPUs            : 1
GPU 00000000:01:01.0
Product Name              : GeForce RTX 3090
Product Brand             : GeForce
```

### 3.72.9 Common Problems

#### Error: enable unsafe interrupts

Add to /etc/default/grub vfio\_iommu\_type1.allow\_unsafe\_interrupts=1

```
GRUB_CMDLINE_LINUX_DEFAULT="vfio_iommu_type1.allow_unsafe_interrupts=1"
```

### Error: iommu group is not viable

If you get this error trying to start the virtual machine with a GPU attached:

```
2021-12-17T07:35:06.533164Z qemu-system-x86_64: -device vfio-pci,host=0000:b1:00.0,
↪id=hostdev0,bus=pci.1,addr=0x1,rombar=1: vfio 0000:b1:00.0:
group 155 is not viable
Please ensure all devices within the iommu_group are bound to their vfio bus driver.
```

This means the PCI you want to pass through has more devices. Possibly an USB embedded is loading its driver and preventing the GPU from being attached in the virtual machine.

```
# dmesg | grep iommu | grep "group 155"
[ 1.893555] pci 0000:b1:00.0: Adding to iommu group 155
[ 1.893653] pci 0000:b1:00.1: Adding to iommu group 155
[ 1.893751] pci 0000:b1:00.2: Adding to iommu group 155
[ 1.893848] pci 0000:b1:00.3: Adding to iommu group 155
```

If your GPU also sports an USB device, you need to check what driver this device is using and blacklist it here also. Use `lspci -Dnn -k` to check what driver it is using. Search for the PCI device numbers you found in the previous command.

```
$ lspci -Dnn -k
000:b1:00.2 USB controller [0c03]: NVIDIA Corporation TU104 USB 3.1 Host Controller_
↪[10de:1ad8] (rev a1)
Subsystem: NVIDIA Corporation TU104 USB 3.1 Host Controller [10de:129f]
Kernel drivers in use: xhci_pci
```

Then add it to the end of the file `/etc/modprobe.d/blacklist-gpu.conf`:

```
blacklist xhci_pci
```

### 3.72.10 References

- <https://mathiashueber.com/windows-virtual-machine-gpu-passthrough-ubuntu/>

## 3.73 Network Bridge

Virtual machines have an internal IP address in a private network by default. They access outside using NAT from the host.

A real IP address can be configured in the virtual machines. It requires a bridge set up in the Ravada server.

KVM provides several ways to configure guest networking, [read this](#) if you want more information. In this document we provide only a guide to configure network bridging and setting this network in the guest virtual machines using netplan.

### 3.73.1 Host Configuration

This is a guide to configure bridge when you host server has the network configured with Netplan. If you have a recent server you probably have it this way. For other ways to configure bridging read the [KVM Networking guide](#).

#### Network interface

**Warning:** The following step will change the host network settings. This may turn the network unusable and the remote access may be disrupted. Make sure you have ways to access the server console just in case you can not access from outside.

First of all you have to identify the network interface. Type `ip a` and find the name of the device after a number, a couple of lines later you will be able to see the IP of your host.

```
2: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group_
↪ default qlen 1000
   link/ether 54:05:db:ac:b1:cb brd ff:ff:ff:ff:ff:ff
   inet 192.168.1.61/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s31f6
```

In this example we can see the name of the network device is `enp0s31f6`.

#### Configure network bridge

Edit the netplan configuration. There you will find the configuration of the network interface we spotted in the previous step. Create a new *bridges* section and add there a new bridge linked to that interface.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s31f6:
      dhcp4: no
  bridges:
    br0:
      interfaces: [enp0s31f6]
      dhcp4: no
      addresses: [192.168.1.106/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [192.168.1.4, 192.168.1.1 , 8.8.8.8]
```

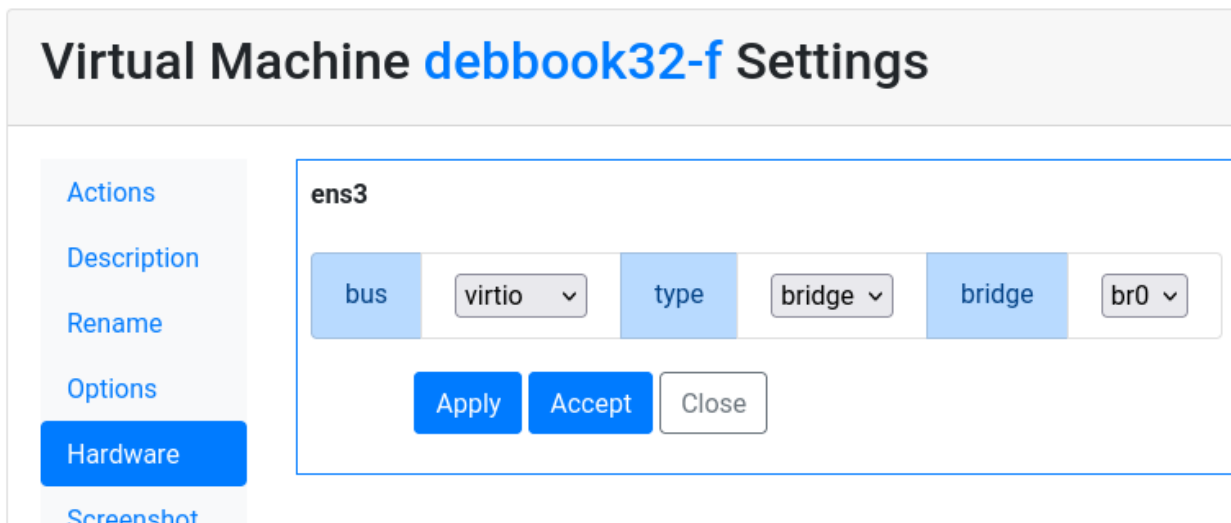
Apply this configuration typing

```
$ sudo netplan try
```

If everything went fine now when you type `ip a` you will see the former interface without any IP assigned, and the new bridge `br0` configured properly. Reboot the server to make sure everything starts correctly.

## Assign the bridge to the virtual machine

Edit the virtual machine hardware network. Set the type to *bridge* and choose the bridge name.



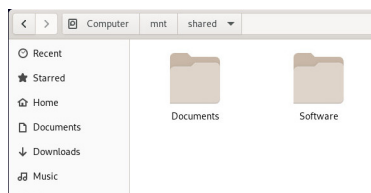
## 3.74 VirtioFS

Virtiofs is a shared file system that lets virtual machines access a directory tree on the host. It is available since Ravada v1.6.

### 3.74.1 Usage Examples

#### Shared Space

Directories and files can be shared from the host to the virtual machines. All kinds of files and data can be stored for read and write access. Read-only mode can be optionally enforced at mount time.



## Private Space

It can be configured a private space for each virtual machine. Unprivileged users will only be able to write and read their own files. It can be granted special access to all the data if it is required. A case usage could be to store students assignments and the teacher is granted access to all the files.

### 3.74.2 Requirements

VirtioFS works only with libvirt 6.2. It is installed in these Linux Distributions:

- Ubuntu 22.04
- Debian 11
- Alpine 3.16

( TODO : Please contribute if you are aware of more , thank you ! )

### 3.74.3 Virtual Machine Configuration

In the hardware section of the virtual machine, add a new *filesystem* item. You have to pass the full path you want to share with the virtual machine.

The screenshot shows a window titled "Virtual Machine **tstbull** Settings". On the left is a sidebar with menu items: "Actions", "Description", "Rename", "Options", "Hardware" (which is highlighted in blue), and "Screenshot". The main area is titled "Add new Filesystem". It contains a label "source" followed by a text input field containing the path "/home/shared". Below the input field are two buttons: "Submit" (in blue) and "Cancel" (in grey).

### 3.74.4 Linux

Using virtiofs from Linux virtual machines is pretty straightforward and the drivers come already with any latest kernel.



## Mount in the Virtual Machine

To mount the partition add a line in the `fstab` with the source name and the directory you want to mount it. In this example we mount the directory exported from `/home/shared`, that will be called `home_shared`. It will be mounted in the path `/mnt/shared` inside the virtual machine.

```
home_shared /mnt/shared virtiofs rw,relatime 0 0
```

Create the mount path `/mnt/shared` and type `mount -a` to try it.

## Mount read-only

In the first example we accessed the directory with read and write options. If you want to access it read only, mount it this way:

```
home_software /mnt/software virtiofs ro,relatime 0 0
```

## 3.74.5 Mount in Windows

See [this manual](#) to use VirtioFS from Windows.

## 3.74.6 Changing the owner of the directory

In the previous section we created a directory for all the virtual machines to see. It is usually ok if you just want to share files read only or that only root can write to. It is possible to allow the user of the desktop to write in this shared location right away.

### Set the user id

Setting the user id is very important because the user will not be able to write unless it is carefully chosen. To do so start the virtual machine and check what is the user id for the user. It is usually created at installation time. If you are not sure what is the name of the user, just boot the base, enter the desktop and that command.

```
user@virtual-machine:~$ id `whoami`
```

The very first numerical value returned is the one you must configure for the shared filesystem.

The owner of the shared filesystem must be set first in the host. Access the server where all the virtual machines are stored and change the owner with the `uid` you just found out in the previous step.

```
frankie@host:~$ sudo chown 1000 /home/shared
```

### Mount Subdirectory in the base

First you have to create the mount point in the virtual machine. The owner of that subdirectory must be the user that runs the desktop.

```
root@virtual-machine:~# mkdir /mnt/shared
root@virtual-machine:~# chown 1000 /mnt/shared
```

Then add it to the `fstab` file so it gets mounted on boot unless you already did in the previous step.

```
home_shared /mnt/shared virtiofs rw,relatime 0 0
```

Reboot and check it is mounted when booting:

```
$ df | grep shared
```

You must see something like this, with the name of the partition first, and the mount point.

```
home_shared /mnt/shared virtiofs rw,relatime 0 0
```

Check the user is able to write there:

```
user@virtual-machine:~$ touch /mnt/shared/test
user@virtual-machine:~$ rm /mnt/shared/test
```

### 3.74.7 Subdirectory Mode

The basic usage is having a directory that all the clones can see. It is possible to have each cloned virtual machine to have its own private space.

This works creating a subdirectory for each clone when it is created.

#### Create Subdirectory filesystem

Create a base and add a shared partition, but this time select the checkbox for *Subdirectory*. Also set the user id that will be the owner of the files.

See the previous step to find out what user id you must define here: “*Changing the owner of the directory*”

## Add new Filesystem

Source ?	<input type="text" value="/home/shared"/>
Subdirectory	<input checked="" type="checkbox"/> <i>Creates a new subdirectory for each clone</i>
User id	<input type="text" value="500"/> <i>User ID for the subdirectory</i>
<div><input type="button" value="Submit"/> <input type="button" value="Cancel"/></div>	

#### Set the owner

Make sure the user is set as the owner of the directory mountpoint,

## Prepare the Base

Once the filesystem is setup and you have checked it boots okay and the user can write there, prepare the base either clicking the checkbox in admin machines or in the virtual machine setup.

## How the clones see it

When the user boots the clone, all that is shown is a directory mounted at /mnt/shared with the contents this user writes.

## How is it stored in the server

If you inspect the contents of the server partition after some clones have used this, you will see something like this:

```
# ls /home/share
drwxr-xr-x 2      1000 root 4096 june  14 11:04 tstbull-username1
drwxr-xr-x 2      1000 root 4096 june  14 11:04 tstbull-username2
```

Further inspections of the content of the subdirectory `tstbull-username1` will provide the files the the user has created from the virtual machine.

## 3.74.8 Read More

More information about VirtioFS.

## 3.75 Firewall

Ravada adds and removes rules to the iptables. If you want to restart the firewall or change some rules you can save the Ravada entries and restore it later.

### 3.75.1 Chains

Ravada has its own chain called *RAVADA* where all the permissions to access the virtual machines displays are stored. You can save it, then apply your own rules, and then restore it.

### 3.75.2 Steps to reload the iptables

#### Step 1: Save the Ravada rules

A line must be added at the beginning and the end of the RAVADA rules so at the end this file can be used in restore.

```
# echo "*filter" > ravada.iptables
# iptables-save | grep RAVADA >> ravada.iptables
# echo COMMIT >> ravada.iptables
```

### Step 2: Apply your own rules

Change your iptables and apply your changes. Now the RAVADA rules will have been flushed.

### Step 3: Restore the Ravada rules

Now we must restore the rules we saved in step 1.

```
# iptables-restore -n < ravada.iptables
```

## 3.76 Best practices to install Linux Guests in Ravada

### 3.76.1 Disk volumes

It is advisable to have at least 2 disk volumes: one for the Operative System and another one for the SWAP partition. Make sure both are selected in the new machine form.

The Operative system will go to /dev/vda and the swap drive will be in /dev/vdb. Configure them properly in the installation process.

#### Installing software

You should at least install these applications:

- qemu-guest-agent
- acpi

### 3.76.2 Energy setup

Configure the system so if the *power off* button is pressed the computer is shut down.

### 3.76.3 Automatic or unattended upgrades

Upgrades may cause the disk volumes to grow unexpectedly. They should be disabled.

For Debian, Ubuntu and similar systems:

```
$ sudo dpkg-reconfigure unattended-upgrades
```

Other Linux flavours may have similar tools. Contributions welcome.

### 3.76.4 Limit systemd log size

Thanks to Jordi Lino for this tip .

You can change the systemd configuration to limit the journal log disk usage (100 MB for example). Edit the `/etc/systemd/journald.conf` file and un-comment (remove # at the beginning) the line `#SystemMaxUse=` and change it to `SystemMaxUse=100M`.

Read [this](#) for more information.

## 3.77 Install Alpine Linux

Alpine is a really small linux distribution. It can be installed in a really tight disk drive with barely no RAM. It is good for testing purposes but it can also be used in production servers.

These are guidelines to install Alpine Linux inside a Ravada KVM Guest.

### 3.77.1 Base Guest

The guest should have more than 256 MB of RAM. If you are planning to run many services you should create the virtual machine with more memory. You can increase it later if you want to keep it slim.

At least 1GB disk drive is required. A swap partition should also be added when creating the virtual machine.

RAVADA VDI Available Machines Admin tools ? Help

### New Machine

From Template From Machine

Backend **i** KVM

Select Template **i** Alpine 3.8 32 bits  
Alpine Linux 3.8 32 bits ( Minimal Linux Distribution )

Select ISO **i** /var/lib/libvirt/images/alpine-standard-3.8.1-x86.iso

Name Langos

Disk Size: (GB) 1 Remove swap The Minimum Disk Size needed for this ISO is 1GB.

Swap Size: (GB) 1

Ram: (GB) 1

Cancel Create

Fig. 5: Create Alpine Virtual Machine

When the machine is created start it from Admin Tools menu, click on Virtual Machines to see a list. At the right there is a bunch of buttons. Click on *view* to start and access the virtual machine console.

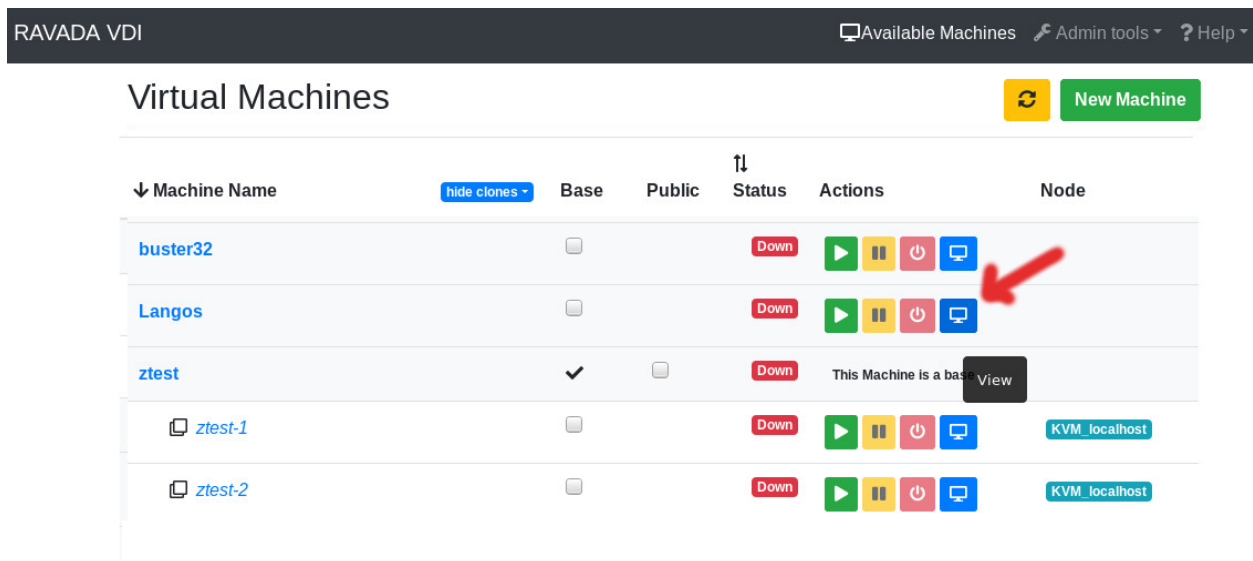


Fig. 6: Start and View Virtual Machine

### 3.77.2 Login

Once it boots login as *root* with no password.

### 3.77.3 Setup

Alpine Linux has a setup script that eases the installation process. These are the default settings we use in a Ravada-KVM Virtual Machine.

Start the setup typing:

```
# setup-alpine
```

If you see an error about missing mirrors you must stop the process pressing CTRL+C , then set manually a mirror like this one:

```
# export MIRRORS=http://mirror.leaseweb.com/alpine/
# setup-alpine
```

## Keyboard

First choose a keyboard layout and variant.

## Hostname

Type a *hostname* for the virtual machine. If you call it with the same name when you created the machine before it would be easier for you:

## Networking

The setup will detect a network interface, probably *eth0*. Accept this value. Also you should probably accept using *dhcp* and answer *no* to *manual network configuration*, unless you know what you are doing.

## Password

Type a new password for the *root* user.

## Timezone

Select a timezone, the default UTC won't do any harm when you are testing, but if you know the actual timezone, type it.

### Proxy

If you are downloading packages through a proxy tell it now, you probably should keep it to *none*.

### Mirror

Choose a mirror for Alpine to download packages from. Type */* if you set the mirror manually with an export before. If not type *f* and it will check your connection and choose the best one, it may take a while.

### SSH Server

Answer *openssh* when asked about an SSH server.

### Partitioning

The setup script will ask the disk

```
Available disks are:
- vda
- vdb
Which disk(s) would you like to use? [none] vda
How would you like to use them ? [?] sys
WARNING: Erase the above disk(s) and continue [y/N]: y
```

Now the setup process will install minimal applications to start the server.

### Reboot

Once it is installed reboot to continue setting this virtual machine.

```
# reboot
```

### Installing more software

You should at least install these applications and start udev:

```
# apk add qemu-guest-agent acpi udev
#
# rc-update add udev
# rc-update add udev-trigger
# rc-update add udev-settle
# rc-update add udev-postmount
# rc-update add qemu-guest-agent
```



### 3.77.4 Shutdown and restart

Type these commands from the console to restart and shutdown the server:

```
# reboot
# poweroff
```

### 3.77.5 Advanced Settings

#### Add a swap partition

If you added a swap volume in the Ravada installation form you must define it later after the Alpine setup script.

#### Define de partition

The swap device will be probably in `/dev/vdb`, check first with `df` it is not mounted already. If it is not shown it is ok.

Using `fdisk` you should find an empty disk and you must create the partitions like this:

```
localhost:~# fdisk /dev/vdb
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-2080, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-2080, default 2080):
Using default value 2080
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table
```

#### Create the swap

```
# mkswap /dev/vdb1
```

#### Test it

Type this to start using the swap space:

```
# swapon -a
```

This command will check the memory utilization, at the bottom there should be now a swap entry:

```
# free
```

	total	used	free	shared	buffers	cached
Mem:	1031924	48896	983028	100	2884	10964
-/+ buffers/cache:		35048	996876			
Swap:	1262140	0	1262140			

## Add it to the system

```
# echo "/dev/vdb1        swap        swap        defaults 0 0" >> /etc/fstab
```

Now the swap space is configured, after rebooting the system it should show typing the *free* command.

## 3.78 Install Windows 10

These are guidelines to install Windows 10 inside a Ravada KVM Guest.

### 3.78.1 Requirements

#### Installation sources

The Windows 10 ISO image file is required:

Download it and copy them in the Ravada host server at the directory `/var/lib/libvirt/images`.

#### Virtual Machine features

The guest should have more than 3 GB of RAM You can increase it later if you want to keep it slim.

At least 30GB disk drive is required. A swap partition should also be added when creating the virtual machine.

### 3.78.2 Create the Virtual Machine

From the web management tools, go to “New Machine” and create select the Windows 10 Template and the ISO image you just downloaded.

It may take a few minutes if it is the first Windows virtual machine that is installed in this system. Ravada will download the additional drivers ISO volume.

The screenshot shows the 'Nova màquina' (New Machine) configuration form in the Ravada VDI web management interface. The form is titled 'Nova màquina' and has two tabs: 'Des d'una plantilla' (Selected) and 'Des d'una màquina'. The form contains the following fields and options:

- Backend:** A dropdown menu set to 'KVM'.
- Escull una plantilla:** A dropdown menu set to 'Windows 10'. Below it, a note says: 'Windows 10 64 bits. Requires an user provided ISO image. [help]'.
- Escull ISO:** A dropdown menu set to 'AvanliblibvirtImages/Win10\_Spanish\_x64.iso'.
- Nom:** A text input field set to 'Win10Slim'.
- Espai de disk (GB):** A slider set to 20. A red button 'Remove swap' and a note 'The Minimum Disk Size needed for this ISO is 21GB.' are visible.
- Espai d'swap (GB):** A slider set to 1.
- Ram (GB):** A slider set to 1.

At the bottom of the form are 'cancel·lar' and 'Crear' buttons. The footer of the interface includes the 'UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH' logo, website and version information (Website: 2.0.0, version: 0.5.0-beta7), license (AGPL v3), and copyright (Copyright © 2016 - 2019 RavadaVDI).

When the machine is created start it from *Admin Tools menu*, click on *Virtual Machines* to see a list. At the right there is a bunch of buttons. Click on *view* to start and access the virtual machine console.



Fig. 7: Start and View Virtual Machine

### 3.78.3 Setup

#### Windows drivers

The installer may not see the virtual machine disk drive. Choosing “Load Drivers” in this screen and select Click on “Examine”, you should see two CD units. The drivers are in the second one called something like “virtio-win”. Choose *amd64* and then select the operative system you are installing.

For *Windows 11* it should be the driver as *Red Hat VirtIO SCSI controller (E:amd64w11viostor.inf)*

Click on *Next* to continue installing Windows.

Follow the usual procedure to install Windows10.

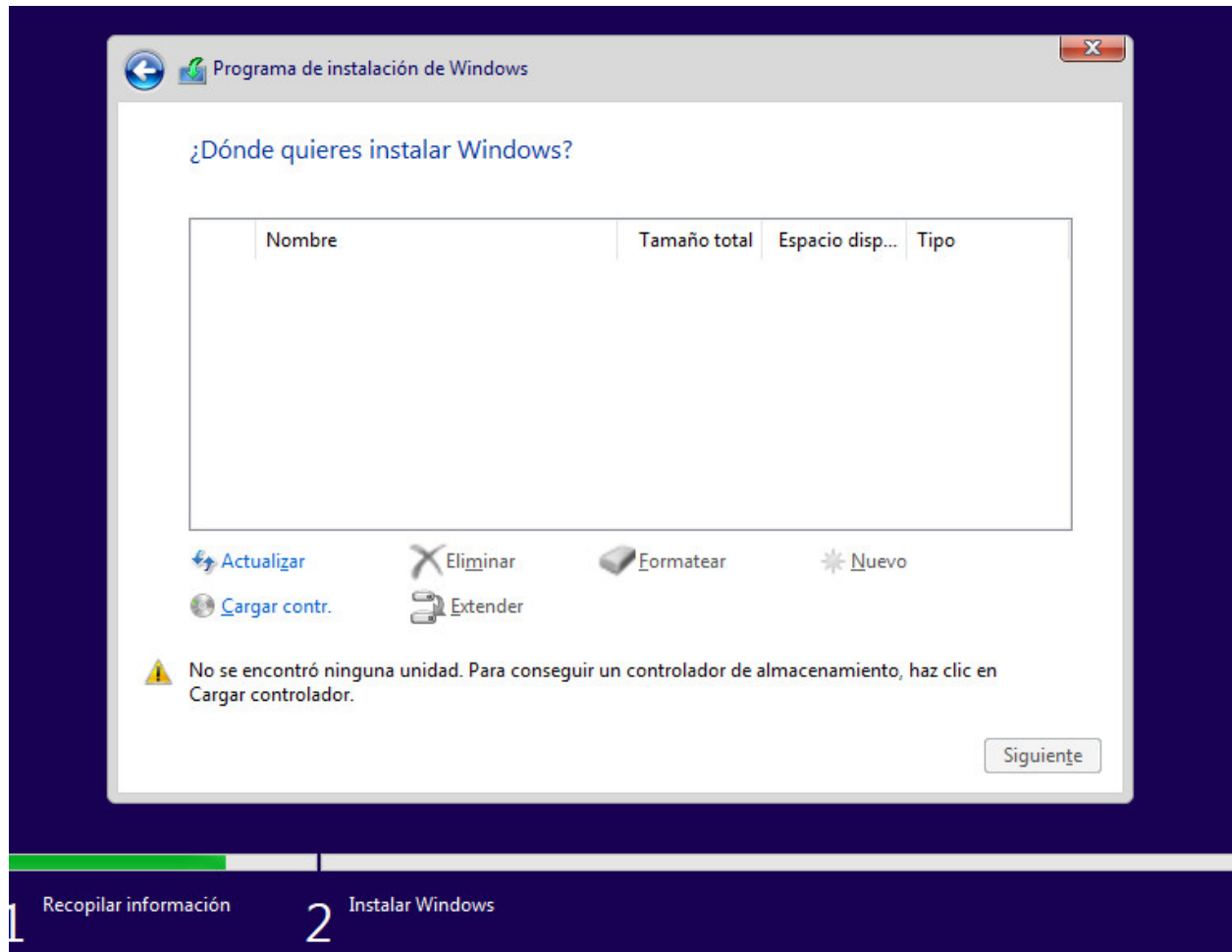
#### Windows Post Install

When the installation is finished, you need to install:

- qemu-guest agent, see the instructions here: <https://pve.proxmox.com/wiki/Qemu-guest-agent#Windows>
- Windows guest tools - [spice-guest-tools](#) .
- make sure that acpi service is activated.

If you experience slow response from the mouse or other glitches you may try installing [VirtIO Drivers](#) .

The drivers CDROM should have been automatically located in the secondary cd drive in your system.



### 3.78.4 Use a swap partition for pagefiles

By default the Windows pagefiles will go to the C: disk drive. That will make the clones partitions grow up too much and fill your host disk drive. It should go to the *SWAP* volume, probably in your D: or E: drive.

In this link you can see how to move pagefiles to another disk:

<https://winaero.com/blog/how-to-move-page-file-in-windows-10-to-another-disk/>

## 3.79 Install Windows 11

These are guidelines to install Windows 11 inside a Ravada KVM Guest.

### 3.79.1 Requirements

Windows 11 requires TPM ( Trusted Platform Module ). Follow [this guide](#)

#### Installation sources

The Windows 11 ISO image file is required:

Download it and copy them in the Ravada host server at the directory `/var/lib/libvirt/images` .

#### Virtual Machine features

The guest should have more than 4 GB of RAM You can increase it later if you want to keep it slim.

At least 60GB disk drive are required. A swap partition should also be added when creating the virtual machine.

### 3.79.2 Create the Virtual Machine

From the web management tools, go to “New Machine” and create select the Windows 11 Template and the ISO image you just downloaded.

It may take a few minutes if it is the first Windows virtual machine that is installed in this system. Ravada will download the additional drivers ISO volume.

When the machine is created start it from *Admin Tools menu*, click on *Virtual Machines* to see a list. At the right there is a bunch of buttons. Click on *view* to start and access the virtual machine console.

### 3.79.3 Install The Operative System

From the Virtual Machines admin tools , click on the blue screen from the one you just created.

The Virtual machine is starting, you will see a link for the spice viewer. Click on it to see the virtual machine screen. The password for the viewer will be copied in the clipboard. Just paste to in the authentication form to access to the viewer.

## New Machine

From Template
From Machine

Backend: KVM

Select Template ⓘ: Windows 11  
Windows 11 64 bits. Requires an user provided ISO image.[\[help\]](#)

Select ISO ⓘ ⓘ: /var/lib/libvirt/images/SW\_DVD9\_Win\_Pro\_11\_21H2\_64BIT\_Spanish\_Pro\_Ent\_EDU\_N\_MLF\_;

Name: test\_windows\_11

Ram: (GB): 4,1





System Disk: (GB): 65 The Minimum Disk Size needed for this ISO is 64GB.

Advanced options ☐

Fig. 8: Set more than 4 GB of RAM and at least 65 GB for the disk drive.

RAVADA VDI					
Màquines disponibles <span>⚙️ Eines d'administrador</span> <span>🔍 Ajuda</span>					
testalp	✓	<input type="checkbox"/>	Aturada	Aquesta màquina és una base	
testbuster	<input type="checkbox"/>		Aturada	▶️ ⏸️ ⏻ 🖥️	
UbuntuPAE	✓	<input type="checkbox"/>	Aturada	Aquesta màquina és una base	
wind10_TracePro_19	<input type="checkbox"/>		Aturada	▶️ ⏸️ ⏻ 🖥️	
windows10_TracePro	✓	<input type="checkbox"/>	Aturada	Aquesta màquina és una base	
Windows10Slim	<input type="checkbox"/>		Aturada	▶️ ⏸️ ⏻ 🖥️	➡️
WindowsEnvirt	✓	✓	Aturada	Aquesta màquina és una base	

Fig. 9: Start and View Virtual Machine

Machine Name	Base	Public	Status	Actions
test_windows_11	<input type="checkbox"/>		Down	   

### Boot from CD

You may see this screen and a *Shell>* prompt.

```

test_windows_11 (1) - Remote Viewer
File View Send key Help

UEFI Interactive Shell v2.2
EDK II
UEFI v2.70 (EDK II, 0x00010000)
Mapping table
  FS0: Alias(s) :CD0a65535a1:;BLK2:
        PciRoot (0x0) /Pci (0x1F,0x2) /Sata (0x0,0xFFFF,0x0) /CDROM (0x1)
  FS1: Alias(s) :F0a65535a:;BLK3:
        PciRoot (0x0) /Pci (0x1F,0x2) /Sata (0x0,0xFFFF,0x0) /UenMedia (C5BD4D42-1A76
-4996-8956-73CDA326CD0A)
  BLK4: Alias(s) :
        PciRoot (0x0) /Pci (0x3,0x0)
  BLK0: Alias(s) :
        PciRoot (0x0) /Pci (0x1F,0x2) /Sata (0x0,0xFFFF,0x0)
  BLK1: Alias(s) :
        PciRoot (0x0) /Pci (0x1F,0x2) /Sata (0x0,0xFFFF,0x0) /CDROM (0x0)
Press ESC in 1 seconds to skip startup.nsh or any other key to continue.
Shell> _

```

If you see this screen is because you missed the CD search screen. Click on *Send Key* at the top of the viewer window, and choose *Ctrl Alt Del*. Click back on the viewer and press a key when you see this message:

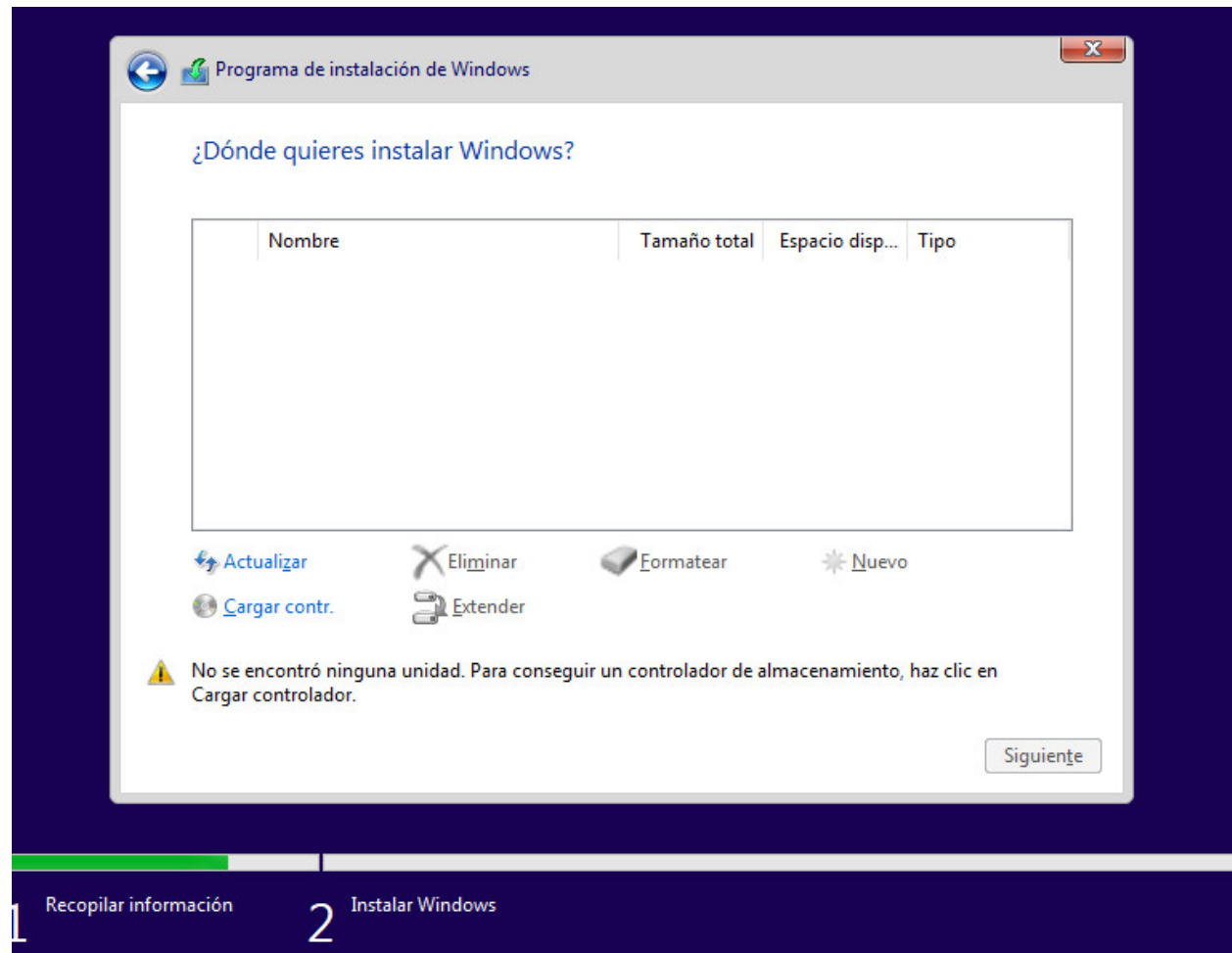
*Press any key to boot from CD or DVD*

## Install Windows 11

Follow the usual procedure to install Windows 11. Select your keyboard, language and other settings.

## Windows drivers

The installer may not see the virtual machine disk drive. Choosing “Load Drivers” in this screen and select



Click on “Examine”, you should see two CD units. The drivers are in the second one called something like “virtio-win”. Choose *amd64* and then select the operative system you are installing.

For *Windows 11* it should be the driver as *Red Hat VirtIO SCSI controller (E:amd64w11viostor.inf)*

Click on *Next* to continue installing Windows.



## Choose Destination Drive

Some disk drives volumes will appear. Choose the bigger one to install Windows 11 there.

### 3.79.4 Setup

Follow the usual procedure to install Windows 11.

When the installations it's finished, you need to install:

## Windows Post Install

When the installations it's finished, you need to install:

- qemu-guest agent, see the instructions here: <https://pve.proxmox.com/wiki/Qemu-guest-agent#Windows>
- Windows guest tools - [spice-guest-tools](#) .
- make sure that acpi service it's activated.

If you experience slow response from the mouse or other glitches you may try installing [VirtIO Drivers](#) .

The drivers CDROM should have been automatically located in the secondary cd drive in your system.

### 3.79.5 Use a swap partition for pagefiles

By default the Windows pagefiles will go to the C: disk drive. That will make the clones partitions grow up too much and fill your host disk drive. It should go to the *SWAP* volume, probabaly in your D: or E: drive.

In this link you can see how to move pagefiles to another disk:

<https://winaero.com/blog/how-to-move-page-file-in-windows-10-to-another-disk/>

## 3.80 How to extend a Ravada Windows guest's disk space

Extending a Windows disk drive in a virtual machine is a straightforward process. Follow this guide carefully.

The process requires execute a hardware change in the Ravada frontend and then use the command line in the host to resize the partition.

### 3.80.1 Shutdown

The virtual machine must be down to resize the volumes. Press *Shutdown* button in the *Admin Tools*.

### 3.80.2 Backup

Make a backup of the disk volumes. The easiest way is to **compact** the virtual machine. After that you should have a copy of all the volumes in the images directory. Usually located at `/var/lib/libvirt/images`.

### 3.80.3 Expand the volume

Go to the *Hardware* tab in the virtual machine settings. Select the disk drive you want to extend and type the desired size of the volume.

Virtual Machine **WindowsE10\_2** Settings

**disk**

WindowsE10\_basic-2-hda.WindowsE10\_basic-hp-hda.qcow2

bus:

capacity:

boot order:

### 3.80.4 Remove and create the partition again

This part of the process must be down in the command line. Connect to the server console and go to the images directory:

```
$ sudo bash
$ cd /var/lib/libvirt/images
```

#### Connect the disk volume as a device

```
root@telecos:/var/lib/libvirt/images# modprobe nbd
root@telecos:/var/lib/libvirt/images# qemu-nbd -c /dev/nbd1 /var/lib/libvirt/images/WindowsE10-hda.qcow2
```

Now the volume appears as an nbd device in the host system. You can use `fdisk` and other tools to change the partitions.

## Remove and create the partition

First let's check what are the partitions with *fdisk*:

```
root@telecos:/var/lib/libvirt/images# fdisk /dev/nbd1
root@telecos:/var/lib/libvirt/images# p
```

```
Disk /dev/nbd1: 110 GiB, 118111600640 bytes, 230686720 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x88e082d8

Device      Boot  Start      End  Sectors  Size Id Type
/dev/nbd1p1 *      2048  1126399  1124352   549M  7 HPFS/NTFS/exFAT
/dev/nbd1p2      1126400 62912511 61786112  29,5G  7 HPFS/NTFS/exFAT
```

The partition we want to change is the second one (nbd1p2). From *fdisk*:

```
# fdisk /dev/nbd1
Command (m for help): d
Partition number (1,2, default 2):
Partition 2 has been deleted.
```

Now we create the partition again but using all the space we just added. *Warning:* when asked about remove the signature, answer N.

```
Command (m for help): n
Partition type
  p   primary (1 primary, 0 extended, 3 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2):
First sector (1126400-230686719, default 1126400):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (1126400-230686719, default 230686719):
Created a new partition 2 of type 'Linux' and of size 109,5 GiB.
Partition #2 contains a ntfs signature.
Do you want to remove the signature? [Y]es/[N]o: N
```

As it was a NTFS partition we change it to that

```
Command (m for help): t
Partition number (1,2, default 2):
Hex code (type L to list all codes): 7
Changed type of partition 'Linux' to 'HPFS/NTFS/exFAT'.
```

Then save and exit *fdisk*:

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

### 3.80.5 Fix the new partition

The new partition must be checked and fixed before resize.

#### ntfsfix

Fix it first in the host:

```
# ntfsfix /dev/nbd1p2
```

#### chkdsk

Now we need to let the Windows virtual machine to check the drive.

First of all disconnect the device from the *nbd* to let it run in the virtual machine. It is a good idea to remove the *nbd* module to make sure it is properly disconnected.

```
# qemu-nbd -d /dev/nbd1  
# rmmod nbd
```

Boot the virtual machine, enter the *cmd* as admin and check the disk. This command will check the disk on the next reboot. Answer yes when asked and reboot the virtual machine. It will probaly warn you that the disk will be checked unless a key is pressed. Do not press any key, let it continue itself. In a few seconds it will be verified.

```
C:> chkdsk c: /f
```

Come back again to the host and tell the filesystem to resize itself to the new full size limit:

```
# modprobe nbd  
# qemu-nbd -c /dev/nbd1 /var/lib/libvirt/images/WindowsE10-hda.qcow2  
# ntfsresize /dev/nbd1p2
```

Disconnect again the nbd and start the virtual machine.

```
# qemu-nbd -d /dev/nbd1  
# rmmod nbd
```

### 3.80.6 Check the new size

Boot the virtual machine again, go to storage properties of the PC. The new size should be available:

#### Local storage



## 3.81 How to extend a Ravada Linux guest's disk space

Extending a Linux disk drive in a virtual machine is a straightforward process. Follow this guide carefully.

The process requires a change in the Ravada frontend and then use the command line in the host to resize the partition.

### 3.81.1 Shutdown

The virtual machine must be down to resize the volumes. Press *Shutdown* button in the *Admin Tools*.

### 3.81.2 Backup

Make a backup of the disk volumes. The easiest way is to **compact** the virtual machine. After that you should have a copy of all the volumes in the images directory. Usually located at `/var/lib/libvirt/images`.

### 3.81.3 Expand the volume

Go to the *Hardware* tab in the virtual machine settings. Select the disk drive you want to extend and type the desired size of the volume.

Virtual Machine **WindowsE10\_2** Settings

Actions  
Description  
Rename  
Options  
Drivers  
**Hardware**  
Screenshot  
Graphics

disk

WindowsE10\_basic-2-hda.WindowsE10\_basic-hp-hda.qcow2

bus

capacity

boot order

### 3.81.4 Remove and create the partition again

This part of the process must be down in the command line. Connect to the server console and go to the images directory:

```
$ sudo bash
```

## Connect the disk volume as a device

```
root@telecos:~# modprobe nbd
root@telecos:~# qemu-nbd -c /dev/nbd1 /var/lib/libvirt/images/linux-user-vda.qcow2
```

Now the volume appears as an nbd device in the host system. You can use `fdisk` and other tools to change the partitions.

## Remove and create the partition

First let's check what are the partitions with `fdisk`:

```
root@telecos:~# fdisk /dev/nbd1
```

```
p
Disk /dev/nbd1: 110 GiB, 118111600640 bytes, 230686720 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x88e082d8

Device      Boot    Start        End    Sectors    Size Id Type
/dev/nbd1p1 *         2048     1050623    1048576    512M  b W95 FAT32
/dev/nbd1p2         2048    1126399    1124352     10G  83 Linux
```

The partition we want to change is the second one (nbd1p2). From `fdisk`:

```
# fdisk /dev/nbd1
Command (m for help): d
Partition number (1,2, default 2):
Partition 2 has been deleted.
```

Now we create the partition again but using all the space we just added. *Warning:* when asked about remove the signature, answer N.

```
Command (m for help): n
Partition type
  p   primary (1 primary, 0 extended, 3 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2):
First sector (1126400-230686719, default 1126400):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (1126400-230686719, default 230686719):
Created a new partition 2 of type 'Linux' and of size 109,5 GiB.
Partition #2 contains an ext4 signature.
Do you want to remove the signature? [Y]es/[N]o: N
```

Then save and exit `fdisk`:

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

### 3.81.5 Fix the new partition

The new partition must be checked and fixed before resize.

Fix it first in the host:

```
# e2fsck /dev/nbd1p2
# resize2fs /dev/nbd1p2
```

### 3.81.6 Start

Disconnect the nbd and start the virtual machine.

```
# qemu-nbd -d /dev/nbd1
# rmmmod nbd
```

Start the virtual machine from the Ravada frontend as usual.

### 3.81.7 Check the new size

Boot the virtual machine again, in a terminal type `df`, it should show the new size.

## 3.82 How to enable KVM virsh console access

### 3.82.1 Requirements

The virtual machine must have the `pty` configured. From the KVM server edit the domain and make sure there is this section:

#### From KVM Server

```
$ sudo virsh edit virtual-machine
```

```
<serial type='pty'>
  <source path='/dev/pts/0'>
  <target type='isa-serial' port='0'>
    <model name='isa-serial'>
  </target>
  <alias name='serial0'>
</serial>
<console type='pty' tty='/dev/pts/0'>
  <source path='/dev/pts/0'>
  <target type='serial' port='0'>
  <alias name='serial0'>
</console>
```

### 3.82.2 From Debian / Ubuntu guest

You either have to enable the serial service or add it to grub.

#### Option 1: Enable Serial Service

```
$ sudo systemctl enable --now serial-getty@ttyS0.service
```

#### Option 2: Add console to grub

Search for the grub.cfg configuration file and add this to `GRUB_CMDLINE_LINUX_DEFAULT`:

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8"
```

### 3.82.3 From KVM server

```
virsh list
Id      Name                                State
-----
1       freebsd                             running
2       ubuntu-box1                         running
3       ubuntu-box2                         running
```

Type the following command from KVM host to login to the guest named ubuntu-box1

```
$ virsh console ubuntu-box1
```

OR

```
$ virsh console 2
```

Use CTRL + 5 to exit the console.

## 3.83 Reduce the image size after cloning a physical PC

Things to keep in mind when we have a cloned image of Windows from a physical PC.

---

**Note:** During these tasks, be aware that it affects the performance of the server. Avoid making them on a Ravada server in production.

---

### 3.83.1 Check the image format

In the following case you can see that it's RAW format. Although the extension of the file is qcow2 this obviously does not affect.

```
qemu-img info Win7.qcow2
image: Win7.qcow2
file format: raw
virtual size: 90G (96636764160 bytes)
disk size: 90G
```



### 3.83.2 STEPS TO FOLLOW

1. Convert from RAW (binary) to QCOW2:

```
# qemu-img convert -p -f raw Win7.qcow2 -O qcow2 Win7-QCOW2.qcow2
```

Now verify that the image format is QCOW2, and it's 26GB smaller.

```
qemu-img info Win7-QCOW2.qcow2
image: Win7-QCOW2.qcow2
file format: qcow2
virtual size: 90G (96636764160 bytes)
disk size: 64G
cluster_size: 65536
Format specific information:
    compat: 1.1
    lazy refcounts: false
    refcount bits: 16
    corrupt: false
```

2. The virt-sparsify command-line tool can be used to make a virtual machine disk (or any disk image) sparse. This is also known as thin-provisioning. Free disk space on the disk image is converted to free space on the host.

```
# virt-sparsify -v Win7-QCOW2.qcow2 Win7-QCOW2-sparsi.qcow2
```

---

**Note:** The virtual machine must be shutdown before using virt-sparsify. In a worst case scenario, virt-sparsify may require up to twice the virtual size of the source disk image. One for the temporary copy and one for the destination image. If you use the `--in-place` option, large amounts of temporary space are not needed.

---

Disk size now is 60G, below you can see that reduce image size in 30GB.

```
qemu-img info Win7-QCOW2-sparsi.qcow2
image: Win7-QCOW2-sparsi.qcow2
file format: qcow2
virtual size: 90G (96636764160 bytes)
disk size: 60G
cluster_size: 65536
Format specific information:
    compat: 1.1
    lazy refcounts: false
    refcount bits: 16
    corrupt: false
```

More information about [https://pve.proxmox.com/wiki/Shrink\\_Qcow2\\_Disk\\_Files](https://pve.proxmox.com/wiki/Shrink_Qcow2_Disk_Files)

Now it is advisable to let Windows do a CHKDSK, do not interrupt it. Finally, you need to install the [Spice guest-tools](#). This improves features of the VM, such as the screen settings, it adjusts automatically, etc.

## 3.84 Qemu Guest Agent

### 3.84.1 Host Qemu Agent Prerequisites

Execute the following commands on your host:

```
$ sudo mkdir -p /var/lib/libvirt/qemu/channel/target
$ sudo chown -R libvirt-qemu:kvm /var/lib/libvirt/qemu/channel
```

And edit the file `/etc/apparmor.d/abstractions/libvirt-qemu` adding the following in the end:

```
$ /var/lib/libvirt/qemu/channel/target/* rw,
```

### 3.84.2 Guest Agent Installation (VM)

This installation must be done in your guest VM if you want to keep the correct time after hibernate.

#### Ubuntu and Debian

```
$ sudo apt install qemu-guest-agent
```

#### Fedora

```
$ dnf install qemu-guest-agent
```

#### RedHat and CentOS

```
$ yum install qemu-guest-agent
```

#### Windows

Follow the instructions provided by [Linux KVM](#)

### 3.84.3 For VM's older than this functionality

If you try to use this function on VM's created before this function was implemented you must do one thing to make it work, first open the machine xml:

```
$ virsh edit <name-or-id-of-your-machine>
```

And add the following inside the 'devices' section:

```
<channel type="unix">
  <source mode="bind"/>
  <target type="virtio" name="org.qemu.guest_agent.0"/>
</channel>
```

That's it, enjoy.

## 3.85 Set Hostname

The hostname of a virtual machine can be changed on startup. The name of the virtual domain is passed in a smbios string and can be used to rename.

### 3.85.1 Requirements

This feature is available from release 0.3.4 and requires libvirt-4.6.

#### Packages

- Ravada: 0.3.4
- libvirt: 4.6

#### Distributions

This feature has been reported to work with these Linux distributions. Any other distribution with libvirt 4.6 or bigger will work too. Please report if you successfully tested it.

Supported distributions:

- Ubuntu 18.10

### 3.85.2 Linux

The virtual machine name can be read with dmidecode

```
$ dmidecode | grep hostname | awk -F: ' print $3'
```

To set the hostname you must create a script that runs on startup, this one line should be enough for most cases:

```
$ hostname `dmidecode | grep hostname | awk -F: ' print $3'`
```

Some tools may read the hostname from the config file, set it like this:

```
$ dmidecode | grep hostname | awk -F: ' print $3' | sed -e 's/^ //' > /etc/hostname
```

#### systemd

If your system supports systemd this script will set the virtual machine name as the hostname on startup. Put the service file in */lib/systemd/system/sethostname.service*:

```
[Install]
WantedBy=multi-user.target

[Unit]
Description=Set Hostname
After=systemd-hostnamed.target syslog.target

[Service]
Type=oneshot
ExecStart=/usr/local/bin/set_hostname.sh
```

This is the script that is launched by the service, it should be in `/usr/local/bin/set_hostname.sh` as specified in the previous file.

```
#!/bin/sh
date >> /var/log/set_hostname.log
hostname=`/usr/sbin/dmidecode | grep hostname | awk '{ print $4}'`
if [ ! -z "$hostname" ]; then
    /bin/hostname $hostname
    /bin/hostname > /etc/hostname
    echo "Found hostname $hostname in dmidecode " >> /var/log/set_hostname.log
else
    echo "Not found hostname in dmidecode " >> /var/log/set_hostname.log
    /usr/sbin/dmidecode >> /var/log/set_hostname.log
fi
```

Type this so the script is executed on startup:

```
$ sudo chmod +x /usr/local/bin/set_hostname.sh
$ sudo systemctl enable sethostname
```

Reboot and check if the hostname is applied. You should find a log file at `/var/log/set_hostname.log`.

### **rc.local**

If you Linux system supports `rc.local` just add this lines to it and the hostname will be updated on boot:

```
hostname `dmidecode | grep hostname | awk -F: '{ print $3}'`
hostname > /etc/hostname
```

## **3.85.3 Windows**

SMBios information is available in Windows too. The data is stored in the registry and also can be shown with a tool called WMI.

Contributed information would be appreciated.

## **3.86 Development Tools**

We're proud to program it in [Perl](#) based on [TDD](#). Perl 5 is a highly capable, feature-rich programming language with over 30 years of development. [More about why we love Perl...](#)

We use [Mojolicious](#), a real-time web framework. We use [MySQL](#). It's the world's most popular open source database. With its proven performance, reliability, and ease-of-use.

We use a lot of powerful free source like [GNU/Linux Ubuntu](#), [KVM](#) or [Spice](#), among others. Responsive web made with [Bootstrap](#) and [AngularJS](#).

We use [Weblate](#) to provide a cleaner and easy to use interface for translators. It's meant for adapting applications and text to enable their usability in a particular cultural or linguistic market.

Then we build documentation and host it in [Read the Docs](#) for you.

## 3.87 Commit Rules

### 3.87.1 Main Branches

The main branches are *master* and *develop* as described here:

<http://nvie.com/posts/a-successful-git-branching-model/>

### 3.87.2 Issues

Please create a new branch for each issue. Also it would be a good idea to call the branch with the number of the issue and a short text, ie:

```
$ git checkout -b 77_crashonstart
```

### 3.87.3 Commit Message

We use conventional commits guideline as specified in <https://conventionalcommits.org/> Each commit must be for a reason, and we should have an issue for that, so we decided to add the issue number in the footer.

Definition:

```
<type>[optional scope]: <description>

[optional body]

footer #issue
```

Example:

```
fix: active virtual machines can not be started

When a virtual machine is already active, do not try to start it and return

#77
```

### 3.87.4 Show the branch in the message

Add the file *prepare-commit-msg* to the directory *.git/hooks/* with this content:

**Note:** Remember to give permission to execute, `chmod a+x prepare-commit-msg`

```
#!/bin/sh
#
# Automatically adds branch name and branch description to every commit message.
#
NAME=$(git branch | grep '*' | sed 's/* //' )
DESCRIPTION=$(git config branch."$NAME".description)
TEXT=$(cat "$1" | sed '/^#.*$/d')
```

(continues on next page)

(continued from previous page)

```
if [ -n "$TEXT" ]
then
    echo "$NAME": '$(cat "$1" | sed '/^#./d')' > "$1"
    if [ -n "$DESCRIPTION" ]
    then
        echo "" >> "$1"
        echo $DESCRIPTION >> "$1"
    fi
else
    echo "Aborting commit due to empty commit message."
    exit 1
fi
```

### 3.87.5 Testing

Before committing, make sure it passes all the tests. This should be run specially when changing the *master* and *develop* branches. Notice some tests require *root* access, so it must run with *sudo*.

```
$ perl Makefile.PL && make && sudo make test
```

If you want to run only one test:

```
$ perl Makefile.PL && make && sudo prove -b t/dir/file.t
```

Proper testing requires the Perl Module Test::SQL::Data, available here: <https://github.com/frankiejol/Test-SQL-Data>

### 3.87.6 Contribution Guide

Check our contribution guide for more information about this topic.

<https://github.com/UPC/ravada/blob/master/CONTRIBUTING.md>

## 3.88 Database Changes

When changing code for this project you may add, remove or modify columns in the SQL database. Those change must be updated too in the *sql* directory.

### 3.88.1 SQL tables

Modify the SQL table definitions in the directory *sql/mysql/*.

### 3.88.2 Data

Some tables may require data, place them in the directory *sql/data*. The files must be called with the *insert* prefix to the table name. So if you create the new table *domaindrivers* you have to :

- Create a file at `sql/mysql/domaindrivers.sql`
- Optionally create a file at `sql/data/insert_domaindrivers.sql` with the insertions

### 3.88.3 SQLite

SQLite definitions are used for testing and are created from the MySQL files. Once the *mysql* file is created, add the new table name to the *sql/mysql/Makefile* and run `make`. It requires <https://github.com/dumblob/mysql2sqlite>

### 3.88.4 Runtime upgrade

When *ravada* runs, it can check if the table definition is accurate. Place the code following the examples at the function *upgrade\_tables* in *Ravada.pm*

Example: To check if the table *vms* has the field *vm\_type*:

```
$self->_upgrade_table('vms','vm_type',"char(20) NOT NULL DEFAULT 'KVM'");
```

## 3.89 Editor configuration rules

If you work with *Ravada* code it will be easier for all of us if you follow a minimum configuration rules.

- expand tabs to spaces
- one tab are 4 spaces

### 3.89.1 Highlight unwanted spaces

Please, don't remove unwanted spaces if aren't yours. Highlight them with these tips: [http://vim.wikia.com/wiki/Highlight\\_unwanted\\_spaces](http://vim.wikia.com/wiki/Highlight_unwanted_spaces)

To disable autoremove of trailing spaces in **Atom**: In Atom Preferences->Packages, select the *whitespace* package. In the *whitespace* package settings, disable "Remove Trailing Whitespace".

### 3.89.2 Vim Example

Set those options in your *.vimrc* to match ours

```
set tabstop=4
set expandtab
"highlight unwanted spaces
highlight ExtraWhitespace ctermbg=red guibg=red
match ExtraWhitespace /\s\+$/
```

## 3.90 Local ISO server

### 3.90.1 ISO Web Server

It is pointless and resource consuming download each time the ISO files from the Internet. Set up a webserver in the main host and let the development virtual ravadas download them from there.

#### Copy the ISO files

Copy the .iso files to the directory /var/www/html/iso.

```
$ sudo mkdir /var/www/html/iso
$ sudo cp /var/lib/libvirt/images/*.iso /var/www/html/iso
```

### 3.90.2 Apache

#### Install Apache

Install apache web server:

```
$ sudo apt-get install apache2
```

#### Config apache

Configure it so ISOs are downloaded from the storage pool, and only the local virtual network is able to access to it.

Edit */etc/apache2/sites-enabled/000-default.conf* and add:

```
<Location /iso>

    Options FollowSymLinks
    AllowOverride None

    Allow from localhost
    Allow from 192.168.122.0/24
    Deny from all

    Require all granted
    Options +Indexes

</Location>
```



## Restart apache

```
$ sudo systemctl restart apache2
```

### 3.90.3 Change the ISO locations

In the table `iso_images` there is an entry that states where are located original ISO files, change it.

#### From localhost

If you want to access to the ISO files from localhost change the *URL* field to this:

```
$ mysql -u root -p ravada
mysql> update iso_images set url = 'http://127.0.0.1/iso/';
```

#### From Virtual Machines

If you install ravada in a virtual machine inside the host you have to change the URLs to the virtual address, it will probably be `192.168.1.1`, check it is doing

```
$ ifconfig virbr0

$ mysql -u root -p ravada
mysql> update iso_images set url = 'http://192.168.122.1/iso/';
```

### 3.90.4 Try it

Remove the ISO from the storage and from the table

#### Remove from the VM storage pool

```
$ sudo rm /var/lib/libvirt/images/*iso
```

#### Remove the device name from the table

First find out the id of the iso image, then remove it.

```
$ mysql -u root -p ravada
mysql> select id,name FROM iso_images;
mysql> update iso_images set device = null where id=9;
```

Restart `rvd_back` and reload the admin page and verify Ravada won't download them from Internet the next time you try to install a new machine.

```
$ sudo ./bin/rvd_back.pl --debug
```

## 3.91 Run Ravada in development mode

Once it is installed, you have to run the two ravada daemons. One is the web frontend and the other one runs as root and manage the virtual machines.

### 3.91.1 Run scripts

Both `rvd_front` and `rvd_back` must run. It is a good practice run each one in a different terminal.

The backend runs as root because it has to deal with the VM processes. It won't reload automatically when there is a change, so it has to be restarted manually when the code is modified:

```
~/src/ravada$ sudo PERL5LIB=./lib ./script/rvd_back --debug
```

The web frontend runs with the `morbo` tool that comes with *mojolicious*. It auto reloads itself if it detects any change in the source code:

```
~/src/ravada$ PERL5LIB=./lib morbo -m development -v ./script/rvd_front
```

### 3.91.2 Stop system Ravada

You may have another copy of Ravada if you installed the package release. **rvd\_back** will complain if it finds there is another daemon running. Stop it with:

```
$ sudo systemctl stop rvd_back; sudo systemctl stop rvd_front
```

### 3.91.3 Run in fish

If you use the fish shell you must run the scripts with these commands:

```
~/src/ravada$ sudo PERL5LIB=./lib script/rvd_back --debug
```

```
~/src/ravada$ set -x PERL5LIB ./lib ; morbo -m development -v script/rvd_front
```

## 3.92 Testing environment

Previously [install](#) TEST::SQL::DATA module.

In project root run:

```
$ perl Makefile.PL
$ sudo make test
```

At the end, in “Test Summary Report” you can check the result.

If something goes wrong you see: Result: FAIL

### 3.92.1 Run a single test

```
$ make; sudo prove -l t/lxc/t
```

### 3.92.2 Advanced Features tests

#### LDAP

Install a [local LDP server](#) to run the LDAP tests.

#### Nodes

Install two virtual machines called ztest-1 and ztest-2 with these features:

- OS: Lubuntu 18.04
- Disk Size: 20 GB
- RAM : At least 4 GB

Follow the remote nodes configuration guide so those machines can be accessed from root in the test host. Also, KVM virtual packages are required. The easiest way is install a virtual machine and clone it twice. Both machines must answer to two IPs as defined in the configurationa.

Place in `t/etc/remote_vm.conf` this config file:

```
ztest-1:
  vm:
    - KVM
    - Void
  host: 192.168.122.151
  public_ip: 192.168.122.251
ztest-2:
  vm:
    - KVM
    - Void
  host: 192.168.122.152
  public_ip: 192.168.122.252
```

#### Base Test machine

Create a small virtual machine called z-test-base:

- OS: Debian Stretch 64 Bits
- Disk: size: 6 GB
- RAM: 1 GB

Configure the [Set Hostname](#) so it gets automatically changed on statup.

You can remove office packages and trim it down with `virt-sparsify`. Install `openssh-server` in base.

Allow root user from host test machine password-less ssh to the PCs.

When everything is set up prepare this machine as a base. When it is done you can run the tests and it will be used to create clones and check stuff on it.

## 3.93 How to create tests

We take great care on crafting a stable product. One of the main keys is creating [automated tests](#) to check everyting works as expected.

As soon as a problem is found, the very first thing is to be able to reproduce it and then create a test case. Make this test fail, so when the code is fixed it should succeed.

### 3.93.1 Test Requirements

Tests run on an blank *sqlite* database that is an exact replica of the real *mysql* database used in production. The fields and data is the same but the data is empty. So you can run the tests in the same host when a real *Ravada* service is running.

To ease the process of creating this *mock* database it is required to install the module [Test-SQL-Data](#) .

### 3.93.2 Test Directory

Create a file in the directory *t* with the *.t* extension. There are subdirs there, try to put the file in one of them.

### 3.93.3 Test File Template

This is an empty tests that does nothing. It just loads a test environment with a blank *sqlite* database. Notice there is a cleaning of the test environment at the begin and end of the tests. This removes possible old leftovers from failed tests.

Let's create a simple test that checks if a virtual machine is removed. So we edit a new file called *30\_remove.t* in the directory *ravada/t*.

```
#!/perl

use strict;
use warnings;
use Test::More;
use Test::SQL::Data;

use lib 't/lib';
use Test::Ravada;

# create the mock database
my $test = Test::SQL::Data->new(config => 't/etc/sql.conf');

# init ravada for testing
init($test->connector);

#####

clean();

use_ok('Ravada');

clean();
```

(continues on next page)

(continued from previous page)

```
done_testing();
```

### 3.93.4 Run

The prove command runs a test file, before you have to prepare the libraries and environment. Tests create remove and manage virtual machines. That requires root access to the system, so sudo should be used to run the tests.

The best thing is to run it this way:

```
$ cd ravada
$ perl Makefile.PL && make && sudo prove -b t/30_remove.t
t/user/30_grant_remove.t .. ok
All tests successful.
Files=1, Tests=1,  2 wallclock secs ( 0.02 usr  0.00 sys +  0.57 cusr  0.10 csys =  0.69
CPU)
Result: PASS
```

### 3.93.5 Trying the Virtual Managers

It is advisable to run the tests on all the virtual managers known to Ravada. To do so we add a loop that tries to load each of them.

Now the test file is like this:

```
#!/perl

use strict;
use warnings;
use Test::More;
use Test::SQL::Data;

use lib 't/lib';
use Test::Ravada;

my $test = Test::SQL::Data->new(config => 't/etc/sql.conf');

init($test->connector);

#####

clean();

use_ok('Ravada');

for my $vm_name ( vm_names() ) {

    my $vm;
    eval { $vm = rvd_back->search_vm($vm_name) };

    SKIP: {
```

(continues on next page)

(continued from previous page)

```

    my $msg = "SKIPPED test: No $vm_name VM found ";
    if ($vm && $vm_name =~ /kvm/i && $>) {
        $msg = "SKIPPED: Test must run as root";
        $vm = undef;
    }

    diag($msg)      if !$vm;
    skip $msg       if !$vm;

    diag("Testing remove on $vm_name");
}
}

clean();

done_testing();

```

We also have a *mock* virtual manager that does nothing but it is used to test generic virtual machines. It is called the *Void* VM and it only should be used for testing. So the output of running the test should be like this:

```

$ perl Makefile.PL && make && sudo prove -b t/30_remove.t
t/user/30_grant_remove.t .. 1/?
# Testing remove on KVM
# Testing remove on Void
t/user/30_grant_remove.t .. ok
All tests successful.

```

### 3.93.6 Test Example: check machine removal

Now the test is there, let's make it check something, like if a virtual machine has been removed.

```

#!/perl

use strict;
use warnings;
use Test::More;
use Test::SQL::Data;

use lib 't/lib';
use Test::Ravada;

my $test = Test::SQL::Data->new(config => 't/etc/sql.conf');

init($test->connector);

#####

sub test_remove {
    my $vm = shift;

    my $domain = create_domain($vm->type);

```

(continues on next page)

(continued from previous page)

```
#    $domain->remove( user_admin );

my $domain2 = $vm->search_domain( $domain->name );
ok(!$domain2,"[".$domain->type."] expecting domain already removed");

}
#####

clean();

use_ok('Ravada');

for my $vm_name ( vm_names() ) {

    my $vm;
    eval { $vm = rvd_back->search_vm($vm_name) };

    SKIP: {
        my $msg = "SKIPPED test: No $vm_name VM found ";
        if ($vm && $vm_name =~ /kvm/i && $>) {
            $msg = "SKIPPED: Test must run as root";
            $vm = undef;
        }

        diag($msg)      if !$vm;
        skip $msg       if !$vm;

        diag("Testing remove on $vm_name");

        test_remove($vm);
    }
}

clean();

done_testing();
```

Now let's run the test:

```
$ perl Makefile.PL && make && sudo prove -b t/30_remove.t
t/user/30_grant_remove.t .. 1/?
# Texting remove on KVM
t/user/30_grant_remove.t .. 3/?
#   Failed test '[KVM] expecting domain already removed'
#   at t/user/30_grant_remove.t line 22.
# Texting remove on Void

#   Failed test '[Void] expecting domain already removed'
#   at t/user/30_grant_remove.t line 22.
# Looks like you failed 2 tests of 7.
t/user/30_grant_remove.t .. Dubious, test returned 2 (wstat 512, 0x200)
Failed 2/7 subtests
```

(continues on next page)

(continued from previous page)

## Test Summary Report

```
-----
t/user/30_grant_remove.t (Wstat: 512 Tests: 7 Failed: 2)
  Failed tests:  4, 7
```

Whoah there ! It looks like the test failed, of course, someone commented the line 19 that actually removes the machine. Uncomment it and run the tests again. It should return OK.

## 3.94 Localization and translation

You can translate Ravada at [Weblate](#). As a feature rich computer aided translation tool, Weblate saves both developers and translators time.



- Automated localization workflow
- Quality checks
- Attribution, all translator are properly credited

You can read weblate features in follow [link](#).

Ravada weblate repository is updated from Github, and the contributions goes automatic to develop branch in Github.

### 3.94.1 New entries

New english entries must be added in the en.po file. It's the origin of the other language files. This new strings will be incorporated automatically in weblate.

**Warning:** Please don't add new entries in other .po files directly. Use [Weblate](#) instead.

The language files are stored [here](#) in lib/Ravada/I18N.

When creating a new translation language, also add it in the frontend so it gets listed for the end users. At around line 1337 in the sub `_translation` add a line like this:

```
sub _translations($c) {
  my %lang_name=(
    ar => 'Arab'
    ,en => 'English'
    ....
    ,XX => 'New language'
```



## 3.95 Ravada Documentation

We have 2 kinds of documents for Ravada : Markdown and reStructuredText.

### 3.95.1 Markdown

Markdown (md) files are the default for github documents. It is used for files stored in the code branches of Ravada.

This is a [guide](#) for the github markdown flavour.

You can view it before commit with the Markdown Preview [Google Chrome extension](#)

### 3.95.2 reStructuredText

reStructuredText (rst) files are used for the advanced documentation of Ravada that you can find in our [readthedocs web site](#)

You can read more about *rst* files at the [docutils rst](#)

The Google Chrome Markdown Preview extension is supposed to render *rst* files too but it lacks some features like remote links. It is better to render those files with [RestView](#).

## 3.96 Frontend Libraries

When upgrading frontend libraries they must be changed in three places: packages.json, scripts and fallback.

### 3.96.1 Packages.json

This file is a MANIFEST and is only used so we receive security alerts from github. Update it with latest updates of the libraries.

### 3.96.2 Scripts

Change scripts pointed in the files `templates/bootstrap/scripts.html.ep` and `templates/bootstrap/header.html.ep`

### 3.96.3 Fallback

Fallback mode can be set up when the libraries are stored in the same Ravada server. A copy of all the libraries is downloaded following the [fallback](#) guide.

Change the fallback files list in the file `etc/fallback.conf`

This file format is:

URL [optional directory/]

So the lines are like this. Notice the first one has the directory and the second line doesn't need one.

`etc/fallback.conf`:

```
https://cdnjs.cloudflare.com/ajax/libs/morris.js/0.5.1/morris.css morris.js/  
https://use.fontawesome.com/releases/v5.10.1/fontawesome-free-5.10.1-web.zip  
...
```

Active the fallback, go to the ravada main source directory and fetch it to check it is working:

### Enable Fallback

Set fallback to 1 in the file `etc/rvd_front.conf`, then restart the frontend.

### Fetch the fallback

```
$ cd ravada  
$ ./etc/get_fallback.pl
```

## 3.97 Hardening Spice security with TLS

TLS support allows to encrypt all/some of the channels Spice uses for its communication. A separate port is used for the encrypted channels.

### 3.97.1 Change libvirt configuration

The certificate must be specified in libvirt configuration file in `/etc/libvirt/qemu.conf`

Uncomment the lines: `spice_listen="0.0.0.0"`, `spice_tls=1` and `spice_tls_x509_cert_dir="/etc/pki/libvirt-spice"`

```
# SPICE is configured to listen on 127.0.0.1 by default.  
# To make it listen on all public interfaces, uncomment  
# this next option.  
#  
# NB, strong recommendation to enable TLS + x509 certificate  
# verification when allowing public access  
#  
spice_listen = "0.0.0.0"  
# Enable use of TLS encryption on the SPICE server.  
#  
# It is necessary to setup CA and issue a server certificate  
# before enabling this.  
#  
spice_tls = 1  
# Use of TLS requires that x509 certificates be issued. The  
# default it to keep them in /etc/pki/libvirt-spice. This directory  
# must contain  
#  
# ca-cert.pem - the CA master certificate  
# server-cert.pem - the server certificate signed with ca-cert.pem  
# server-key.pem - the server private key  
#  
# This option allows the certificate directory to be changed.
```

(continues on next page)

(continued from previous page)

```
#
spice_tls_x509_cert_dir = "/etc/pki/libvirt-spice"
```

### 3.97.2 Add path in Apparmor

You may want to add this path to Apparmor, in some Linux distributions it is not necessary, ie Ubuntu from 18.04.

Add `/etc/pki/libvirt-spice/** r`, in `/etc/apparmor.d/abstractions/libvirt-qemu`

```
# access PKI infrastructure
/etc/pki/libvirt-vnc/** r,
/etc/pki/libvirt-spice/** r,
```

**Note:** Remember restart the services: `systemctl restart apparmor.service & systemctl restart libvirtd.service`

### 3.97.3 Create self signed certificate

Download and run the `create_cert.sh` script.

```
#!/bin/bash

SERVER_IP="$1"
if [ -z "$SERVER_IP" ]; then
    echo "Error, server ip required."
    echo " Usage: $0 ip"
    exit -1
fi

# change the next line
SUBJECT="/C=IL/L=Raana/O=Red Hat"

SERVER_KEY=server-key.pem

# creating a key for our ca
if [ ! -e ca-key.pem ]; then
    openssl genrsa -aes256 -out ca-key.pem 2048
fi
# creating a ca
if [ ! -e ca-cert.pem ]; then
    openssl req -new -x509 -days 1095 -key ca-key.pem -out ca-cert.pem \
        -subj "${SUBJECT}/CN=my CA"
fi
# create server key
if [ ! -e $SERVER_KEY ]; then
    openssl genrsa -out $SERVER_KEY
fi
# create a certificate signing request (csr)
if [ ! -e server-key.csr ]; then
    openssl req -new -key $SERVER_KEY -out server-key.csr -subj "$SUBJECT/CN=$SERVER_IP"
```

```
fi
# signing our server certificate with this ca
if [ ! -e server-cert.pem ]; then
    openssl x509 -req -days 1095 -in server-key.csr -CA ca-cert.pem -CAkey ca-key.pem
    ↪-set_serial 01 -out server-cert.pem
fi

# now create a key that doesn't require a passphrase
openssl rsa -in $SERVER_KEY -out $SERVER_KEY.insecure
mv $SERVER_KEY $SERVER_KEY.secure
mv $SERVER_KEY.insecure $SERVER_KEY

# copy *.pem file to /etc/pki/libvirt-spice
if [ ! -d "/etc/pki/libvirt-spice" ]
then
    mkdir -p /etc/pki/libvirt-spice
fi
cp ./*.pem /etc/pki/libvirt-spice
chown :kvm /etc/pki/libvirt-spice/*.pem
chmod g+rx /etc/pki/libvirt-spice/*.pem

# echo --host-subject
echo "your --host-subject is" `openssl x509 -noout -text -in server-cert.pem | grep
↪Subject: | cut -f 10- -d " "`
```

**Warning:** Whatever method you use to generate the certificate and key files, the Common Name value used for the server and client certificates/keys must each differ from the Common Name value used for the CA certificate. Otherwise, the certificate and key files will not work for servers compiled using OpenSSL.

### 3.97.4 Disable Spice Password

More information about [removing SPICE password](#) for all the networks.