Kernel-based Virtual Machine (KVM) Plugin Plugin

Bacula Systems Documentation

Contents

1 Scope 2

2 Features 3

3 Installation 4

4 Configuration 6

5 Operations 12

6 Limitations 16

Contents

Important: Remember to read the Best Practices chapter common for all of our hypervisor plugins.

This document aims at presenting the reader with information about the **Bacula Enterprise Kernel-based Virtual Machine (KVM) Plugin**. The document briefly describes the target technology of the plugin, defines the scope of its operations, and presents its main features, including clusters.

1 Scope

Bacula Enterprise KVM module is supported on Red Hat Enterprise Linux. The KVM Plugin is designed to be used when the hypervisor uses local storage for virtual machine disks and **libvirtd** for virtual machine management.

The current version of the KVM Plugin is compatible with:

- Proxmox (note that as of **Bacula Enterprise** 10, a specific Proxmox Plugin is available)
- Redhat RHV or Ovirt
- OpenStack

Kernel-based virtual machine (KVM) is an open source virtualization infrastructure built for Linux OS. It requires x86-based processor architecture to operate. KVM consists of two technology components: kernel and user-space. Kernel has two loadable modules: kvm.ko, and either kvm-intel.ko or kvm-amd.ko. The kvm.ko module is responsible for core architecture-independent virtualization processing. The latter two correspond to Intel and AMD processor-specific modules. These modules were merged into the Linux kernel as of version 2.6.20.

See also:

- Go to *KVM Features*
- Go to *KVM Installation*
- Go to *KVM Configuration*
- Go to *KVM Operations*
- Go to *KVM Limitations*

Go back to *the main KVM Plugin page*.

Go back to the main Dedicated Backup Solution page.

2 Features

Bacula Enterprise is an especially secure, scalable and modular backup and recovery software solution with a wide range of capabilities. Backup up KVM hypervisor(s) and its data in a comprehensive manner is one of its many qualities.

Bacula System's KVM Module is fast, reliable, especially simple to use and establishes confidence in the backup system administrator.

The **Bacula Enterprise** KVM Plugin provides the following main features:

- Automatic virtual machine discovery
- File level backup of virtual machines
- Full, Differential, and Incremental backup level support
- The ability to handle inclusion/exclusion of files
- Backup of virtual machines in a 'running', 'paused' or 'shut off' state

KVM enables Linux to do the system grunt work, so it can focus on handling new virtualization instructions exposed by the hardware. Another key benefit for KVM is that it inherits any continuing system improvement from upstream in the larger Linux community.

It is crucial that the kernel modules do not emulate the virtual machine hardware that the guest OS runs on. KVM uses QEMU to build the virtual machines (VM) that interact with guest OS. Each VM is a regular Linux process, which means familiar Linux commands like 'top' and 'kill' can be used to manage and monitor the VMs.

KVM transforms Linux into a type 1 hypervisor, also known as a bare-metal hypervisor. Type 1 hypervisors directly access the computer's hardware and replace the host OS. They are also considered secure because there is nothing between a hypervisor and a CPU that a potential attacker could tamper with. As part of the Linux kernel, KVM has all the OS-level components. This means that each VM works like a regular Linux process and has dedicated virtual components like memory, disks, CPU and network card.

Bacula Enterprise data backup and recovery allows you to back up not only KVM environments, but covers your entire IT estate through a single plane of glass. It brings unprecedented levels of security, business value benefits and low cost of ownership.

See also:

- Go back to *KVM Scope*
- Go to *KVM Installation*
- Go to *KVM Configuration*
- Go to *KVM Operations*
- Go to *KVM Limitations*

Go back to *the main KVM Plugin page*.

Go back to the main Dedicated Backup Solution page.

3 Installation

This article describes how to install Bacula Enterprise Kernel-based Virtual Machine (KVM) Plugin.

3.1 KVM Installation with BIM

In order to install the KVM Plugin with BIM, install the File Daemon with BIM and choose to install the KVM Plugin during the FD installation.

Click here for more details on the plugin installation process with BIM.

See also:

See an alternative way of installing the KVM Plugin - *KVM Installation with Package Manager*.

Go back to the *main KVM Plugin Installation page*.

Go back to the *main KVM Plugin page*.

3.2 KVM Installation with Package Manager

Packages of the KVM Plugin are available for supported platforms. Make a request from your Customer Portal to access the KVM Plugin. The KVM Plugin package needs to be installed onot the KVM host server (hypervisor host), on which **Bacula File Daemon** is already installed.

Note: The KVM Plugin uses snapshots while backing up guest VMs. During a snapshot, blocks modified by the guest VM need to be copied in temporary space, the space required depends on the guest disk activity. By default the space is allocated under `/var/tmp`.

See the articles for specific operating systems:

Installation with Package Manager on Debian/Ubuntu

1. Configure the package manager:

Add the following to a file `/etc/apt/sources.list.d/bacula.list`

```
# Bacula Enterprise - KVM
deb https://www.baculasystems.com/dl/@@customer@@/debs/kvm/@@bee-version@@/@@os-
↪version@@-@@arch@@/ @@os-version@@ kvm

**@@bee-version@@** should be replaced by the version of Bacula
Enterprise you purchased (16.x.y, 14.x.y)

**@@os-version@@** is the code name of the distribution
(buster/stretch/jessie)

**@@arch@@** Architecture: 32 or 64 bit
```

Note: On Ubuntu 64 bit systems you will need to write **deb [arch=amd64]** instead of **deb**.

A complete example might look like this:

Debian:

```
# Bacula Enterprise
deb https://www.baculasystems.com/dl/Customer-123456/debs/kvm/16.0.7/buster-64/ buster kvm
```

Ubuntu:

```
# Bacula Enterprise
deb https://www.baculasystems.com/dl/Customer-123456/debs/kvm/16.0.7/bionic-64/ bionic kvm
```

2. Update your package manager and verify your **Bacula Enterprise** repositories are correctly configured.

```
apt-get update
```

3. Run this command to install the **Bacula Enterprise** packages:

```
apt-get install bacula-enterprise-kvm-plugin
```

See also:

- *Installation with Package Manager on RHEL/CentOS*

Go back to the *KVM Installation with Package Manager*

Installation with Package Manager on RHEL/CentOS

1. Configure the package manager:

Add the following to a file `/etc/yum.repos.d/bacula.repo`

```
#  [Bacula-Enterprise-KVM]
name= Bacula Enterprise KVM
baseurl=https://www.baculasystems.com/dl/@@customer@@/rpms/kvm/@@bee-version@@/@@rhel@@-@@arch@@/
enabled=1
protect=0
gpgcheck=1

**@@bee-version@@** should be replaced by the version of Bacula
Enterprise you purchased (16.x.y, 14.x.y)

**@@rhel@@** is the version of your RedHat/CentOS distribution (9/8/7)

**@@arch@@** Architecture: 32 or 64 bit
```

A complete example might look like this:

```
[Bacula-Enterprise]
name=Red Hat Enterprise - Bacula - Enterprise
baseurl=https://www.baculasystems.com/dl/Customer-123456/rpms/kvm/16.0.7/rhel7-64/
enabled=1
protect=0
gpgcheck=1
```

2. Update your package manager and verify your **Bacula Enterprise** repositories are correctly configured.

```
yum update
```

3. Run this command to install the **Bacula Enterprise** packages:

```
yum install bacula-enterprise-kvm-plugin
```

See also:

- *Installation with Package Manager on Debian/Ubuntu*

Go back to the *KVM Installation with Package Manager*

Go back to the *KVM Installation page*.

Go back to the *main KVM Plugin page*.

See also:

- Go back to *KVM Scope*
- Go back to *KVM Features*
- Go to *KVM Configuration*
- Go to *KVM Operations*
- Go to *KVM Limitations*

Go back to *the main KVM Plugin page*.

Go back to the main Dedicated Backup Solution page.

4 Configuration

This article describes how to configure Bacula Enterprise KVM Plugin.

4.1 File Daemon Configuration

On the KVM host server, the **Plugin Directory** option of the **File Daemon** resource in `/opt/bacula/etc/bacula-fd.conf` has to point to where the `kvm-fd.so` plugin is installed. The standard directory for Bacula plugins is `/opt/bacula/plugins`

```
FileDaemon {
   Name = bacula-fd
   Plugin Directory = /opt/bacula/plugins
   ...
}
```

The `libvirtd` daemon should be started and accessible. The KVM host server's File Daemon should have local, network-mounted, or SAN access to where KVM images are stored. On the KVM host server, the following command should list all local VMs:

```
# virsh list --all
 Id    Name                           State
----------------------------------------------------
```

(continues on next page)

(continued from previous page)

```
 1     gentoo                         running
 2     centos                         paused
 -     debian                         shut off
```

If specification of a URI using the virsh `-c` parameter is required, the corresponding **uri=** parameter will also be needed in the plugin command.

```
# virsh -c qemu:///system list --all
 Id    Name                           State
----------------------------------------------------
 1     gentoo                         running
 2     centos                         paused
 -     debian                         shut off
```

The `install-kvm.sh` script is designed to test the KVM setup of the hypervisor. The script should report an “OK” at the end. If not, any problems reported need to be corrected, and the resulting output should be sent to the Bacula Systems support team if you need assistance.

```
# /opt/bacula/scripts/install-kvm.sh check
Enter the :term:`libvirt` URI to connect libvirtd [qemu:///system]:

Trying to list VMs using virsh -r -c 'qemu:///system' list --all, it should not ask for a password.
Id    Name                           State
----------------------------------------------------
 1     gentoo                         running
 2     centos                         paused
 -     debian                         shut off

Did you have to enter a password to get the VM list? [y/N]: N

Enter the name of a guest that will be used to test the :term:`KVM` plugin requirements:
gentoo

Trying to mount gentoo filesystem as /tmp/bee-kvm-gentoo.2vsYz
Mount OK.
Attempting to list 10 files from gentoo root filesystem.
/tmp/bee-kvm-gentoo.2vsYz/
/tmp/bee-kvm-gentoo.2vsYz/bin
/tmp/bee-kvm-gentoo.2vsYz/bin/bb
/tmp/bee-kvm-gentoo.2vsYz/bin/dd
/tmp/bee-kvm-gentoo.2vsYz/bin/cp
/tmp/bee-kvm-gentoo.2vsYz/bin/df
/tmp/bee-kvm-gentoo.2vsYz/bin/du
/tmp/bee-kvm-gentoo.2vsYz/bin/ip
/tmp/bee-kvm-gentoo.2vsYz/bin/ln
/tmp/bee-kvm-gentoo.2vsYz/bin/ls

Unmounting gentoo filesystem.
OK: All tests are good.
```

See also:

- Go to *Director Configuration*

- Go to *Fileset Examples*

Go back to *the main KVM Configuration page*.

Go back to *the main KVM Plugin page*.

4.2 Director Configuration

KVM Plugin Options

Table 1: KVM Plugin Parameters

Option	Default	Description
uri		The URI parameter specifies how to connect to the hypervisor. The documentation page at http://libvirt.org/uri.html list the values supported.
host		Virtual Machine to be backed up. It is possible to specify a list of of hosts separated by a ‘,’ (without spaces).
include		Specify files or directories to backup. It is possible to specify multiple include= parameters in the plugin command line. See <includeexclude>
exclude		Specify files or directories to exclude. It is possible to specify multiple exclude= parameters in the plugin command line. See <includeexclude>
host_prefix	No	Prefix all files with the virtual machine name (mandatory when doing multiple virtual machine backup in the same job).
host_sep	,	Specify a host separator used in the host parameter. Example: `host_sep=:`, `host=h1:h2:h3`
abort_on_error	No	Abort the job if an error is occurring during the job. By default, if a VM is not accessible for example, the Job will end with JobStatus OK (T) and some JobErrors.
timeout	300 seconds	Number of seconds used to run various commands or wait to connect KVM during during a backup (starting with Bacula Enterprise 12.6.1).
unix_user		Unix user used to execute KVM commands via sudo (starting with Bacula Enterprise 12.6.1).

See also:

- Go back to *File Daemon Configuration*
- Go to *Fileset Examples*

Go back to *the main KVM Configuration page*.

Go back to *the main KVM Plugin page*.

4.3 Fileset Examples

The FileSet example below will backup all files from both the centos and gentoo virtual machines.

```
FileSet {
  Name = FS_KVM
  Include {
    Options {
      Signature = MD5
      Compression = LZO
```

(continues on next page)

(continued from previous page)

```
    }
    Plugin = "kvm: host=centos,gentoo"
  }
}
```

If a KVM FileSet contains multiple virtual machines, each file's path in the Catalog will begin with the name of the virtual machine as shown in the example listing below.

```
* list files jobid=100
+---------------------------------+
| filename                        |
+---------------------------------+
| ...                             |
| /centos/boot                    |
| /centos/boot/grub               |
| /centos/boot/grub/menu.lst      |
| ...                             |
| /gentoo/etc/passwd              |
| /gentoo/etc/group               |
| /gentoo/etc/hosts               |
| ...                             |
+ --------------------------------+
```

If a KVM FileSet contains only one virtual machine, each file's path will not be prefixed with the virtual machine's name, as shown below. This default behavior can be overridden by using the **host_prefix** KVM plugin parameter listed in table *KVM Plugin Options*.

```
FileSet {
  Name = FS_KVM_centos
  Include {
    Plugin = "kvm: host=centos"
  }
}
```

```
* list files jobid=101
+---------------------------------+
| filename                        |
+---------------------------------+
| ...                             |
| /boot                           |
| /boot/grub                      |
| /boot/grub/menu.lst             |
| ...                             |
+ --------------------------------+
```

Including and Excluding Files

With the KVM Plugin, by default all files in the virtual machine are backed up and the FileSet's Include and Exclude directives are ignored.

```
FileSet {
  Name = FS_KVM_broken
  Include {
    Plugin = "kvm: host=centos"  # all files on centos VM will be backed up
    File = /etc                  # will be ignored due to KVM plugin
    File = /home                 # will be ignored due to KVM plugin
  }
  Exclude {
    File = /tmp                  # will be ignored due to KVM plugin
  }
}
```

Note: Selection rules defined in Options blocks are still applied.

If certain directories should be excluded from the virtual machine's backup, the **include** and **exclude** plugin options can be used. The FileSet example below will backup files under `/etc` and `/home`. Files below `/home/tmp` will be explicitly excluded.

```
FileSet {
  Name = FS_KVM_etc_home
  Include {
    Plugin = "kvm: host=centos include=/etc include=/home exclude=/home/tmp"
  }
}
```

When the **include** plugin option is specified, the default behavior of backing up all files in the VM is overridden and **only** directories specified by the **include** plugin options are backed up.

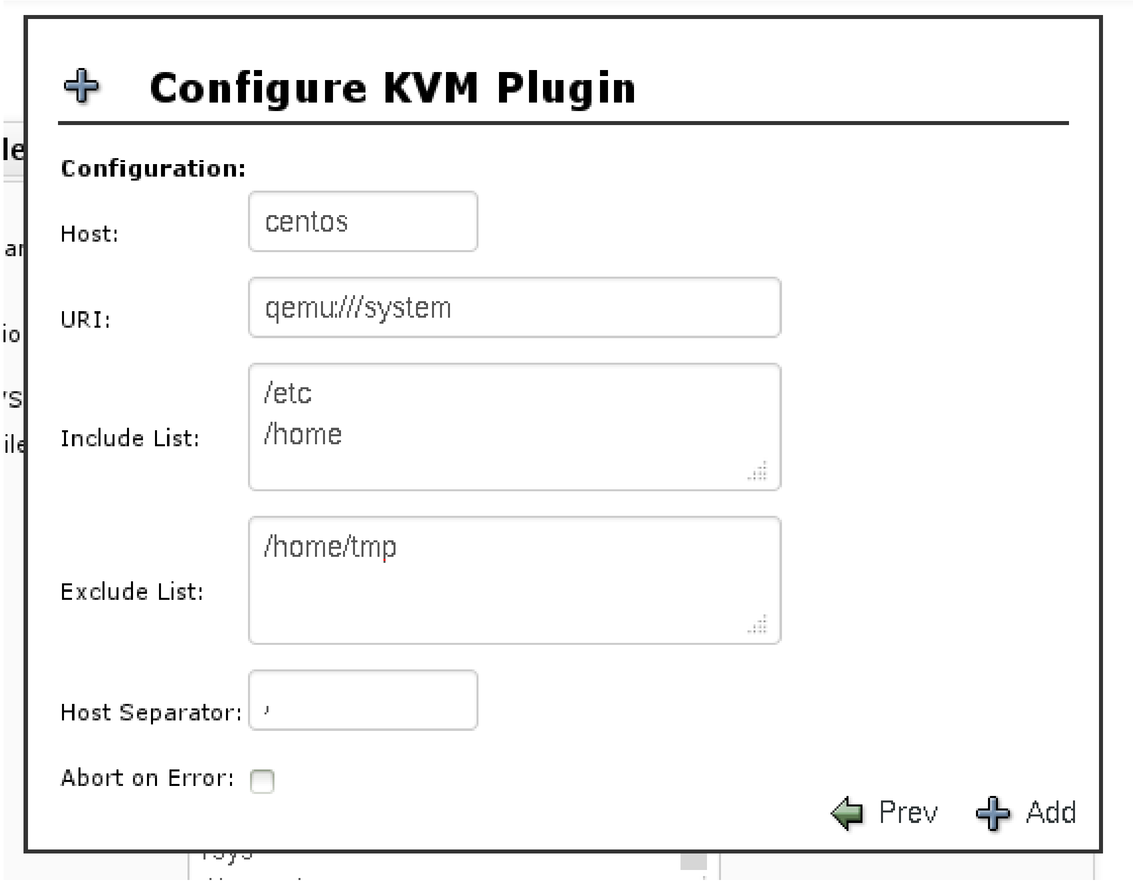
To manage exclude lists, it is also possible to use the **Exclude Dir Containing** FileSet directive. In the example below, all files and directories on the virtual machine named “centos” will be backed up except for directories containing a file named `.excludeme`

```
FileSet {
  Name = FS_KVM_centos
  Include {
    Options {
    Signature = MD5
    }
    Exclude Dir Containing = ".excludeme"
    Plugin = "kvm: host=centos"
  }
}
```

If more complex Include or Exclude capabilities are required, it may be useful to consider installing a Bacula File Daemon inside the virtual machine instead of using the KVM Plugin.

See also:

- Go back to *Director Configuration*



Configure KVM Plugin

Configuration:

Host: centos

URI: qemu:///system

Include List:
/etc
/home

Exclude List:
/home/tmp

Host Separator: ,

Abort on Error: □

Fig. 1: KVM Configuration

- Go back to *File Daemon Configuration*

Go back to *the main KVM Configuration page*.

Go back to *the main KVM Plugin page*.

See also:

- Go back to *KVM Scope*
- Go back to *KVM Features*
- Go back to *KVM Installation*
- Go to *KVM Operations*
- Go to *KVM Limitations*

Go back to *the main KVM Plugin page*.

Go back to the main Dedicated Backup Solution page.

5 Operations

This article describes details regarding backup and restore with Bacula Enterprise KVM Plugin.

5.1 Backup

Common Approaches

To prevent data loss and facilitate system recovery in case the VM is compromised, it is crucial to back up KVM. A common backup strategy is to use a custom script like perl or something else. This type of script can be flexible and can be configured to suit the setup and backup needs of the VM. This process generally involves identifying the VM and for each VM, taking a snapshot of the disks. The snapshot is then dumped on a remote server or a backup storage.

Some scripts may require temporary suspension of the VM's operations. A risk to using scripts is that, while it is easy to access and execute, it can also open up VMs to security vulnerabilities.

Another way to use the snapshot tool is to install all the necessary components on the VM and use this as a base snapshot when it is time to restore.

A backup software may be installed on each VM. The backup process itself can affect the performance of the VM on the server.

This module removes the need to have a client installed in each VM. The module works at the Hypervisor Level, in the same way as Bacula's XEN and PROXMOX modules. The module is a fully integrated, agentless technology.

To back up, the user simply defines a file set.

However, in the instance of needing to restore a file to a running VM, the user would need to have a Bacula Enterprise client installed.

Backup with Bacula Enterprise

Bacula Enterprise enables easy backup and restoration of KVMs at a file level. There is no software installation and scripting required for individual clients. Bacula Enterprise's high level of granularity allows users a high level of control over the data they need to (or don't need to) backup and recover. Auto-detection capabilities also enable automatic VM discovery, which means there is no need to define which VM to backup after initial configuration.

With Bacula Enterprise, it is possible to conduct KVM backup whether the VM is running, paused or shut off. Backing up VMs is a resource-intensive process that can reduce resource allocation to the OS, application files, user data and settings, and consequently, affect the performance of the VM. Bacula's backup tool enables KVM backup without service interruption and data consistency.

Snapshot backup is also available. When this is combined with Bacula Systems' Global Endpoint Deduplication engine and comm-line compression, users can save storage and reduce use of network bandwidth.

See also:

- Go to *KVM Restore*

Go back to *the main Operations page*.

Go back to *the main KVM Plugin page*.

5.2 Restore

Virtual machines are backed up at the file level. To restore a set of files to a virtual machine, a Bacula File Daemon must be installed inside the virtual machine. It is also possible to restore files to a different Bacula client and copy the files to the intended virtual machine afterward.

To restore a file, simply use the bconsole `restore` command, select the backup Job and run a traditional restore Job.

The simplicity of this module can be of great benefit to a System Administrator. Imagine a scenario where we have a "hacked" server that is sending DDOS to the network, and the Sys Admin needs a file from it. She does not want to put her organization's network in danger or at risk by starting a VM to restore a file. Instead, she only selects the specific file to restore, without the need to restore the full VM.

Restoring to a Different Bacula Client

It is also possible to restore files to a different Bacula client and copy the files to the intended virtual machine afterward. Some of the advantages of being able to restore a single VM guest file instead the whole VM are:

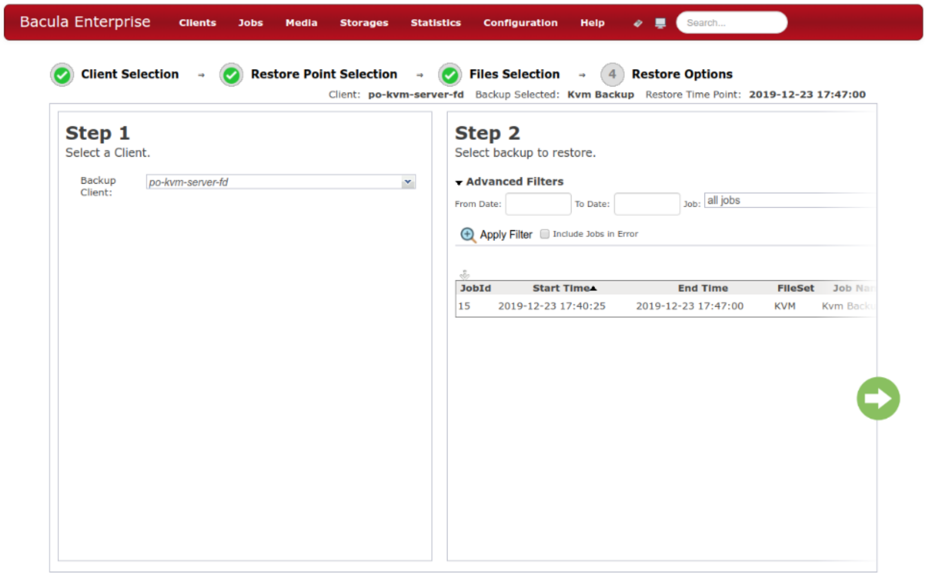
- Saves Time, Network Bandwidth, and Hard Disk Space
- Security
- Facilitates the Restore Workflow
- Removes the need to stop or start a new virtual machine

Single File Restore

Single File Restore is possible with all VM guests where the "libguestfs" library is compatible. KVM and Bacula's KVM module use the library to access the guest's file system.

Restoring with BWeb

The images below demonstrate the sequence to recover KVM with Bacula's BWeb GUI interface



Client: **po-kvm-server-fd** Backup Selected: **Kvm Backup** Restore Time Point: **2019-12-23 17:47:00**

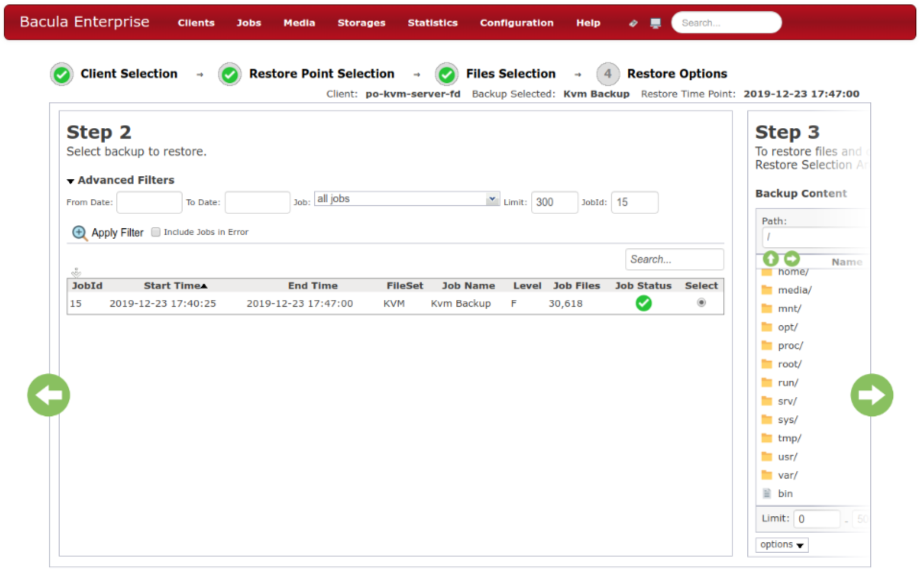
Step 1
Select a Client.

Backup Client: po-kvm-server-fd

Step 2
Select backup to restore.

Advanced Filters
From Date: To Date: Job: all jobs
□ Include Jobs in Error

JobId	Start Time▲	End Time	FileSet	Job Na
15	2019-12-23 17:40:25	2019-12-23 17:47:00	KVM	Kvm Back



Client: **po-kvm-server-fd** Backup Selected: **Kvm Backup** Restore Time Point: **2019-12-23 17:47:00**

Step 2
Select backup to restore.

Advanced Filters
From Date: To Date: Job: all jobs Limit: 300 JobId: 15
□ Include Jobs in Error

JobId	Start Time▲	End Time	FileSet	Job Name	Level	Job Files	Job Status	Select
15	2019-12-23 17:40:25	2019-12-23 17:47:00	KVM	Kvm Backup	F	30,618	✓	●

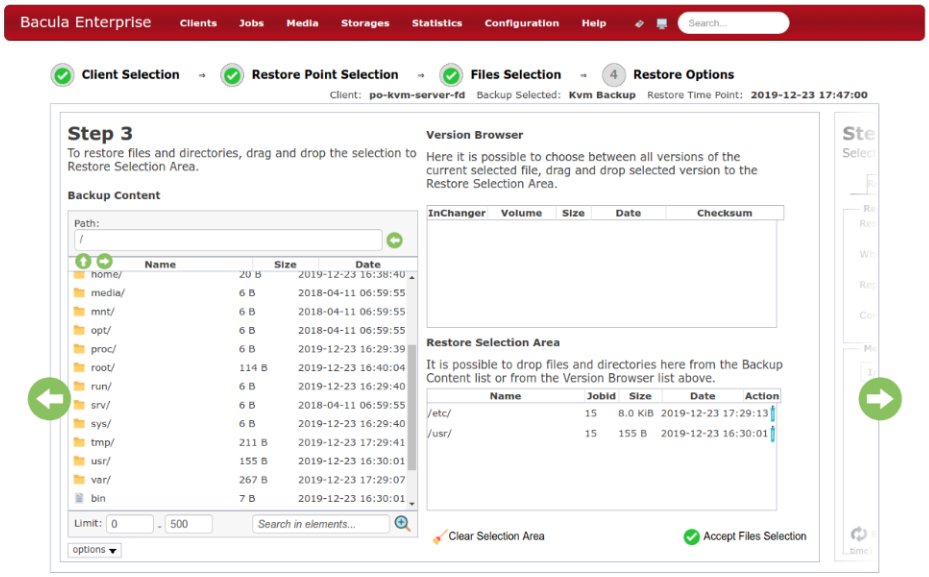
Step 3
To restore files and
Restore Selection A

Backup Content

Path: /

Name
home/
media/
mnt/
opt/
proc/
root/
run/
srv/
sys/
tmp/
usr/
var/
bin

Limit: 0 - 50



Client: **po-kvm-server-fd** Backup Selected: **Kvm Backup** Restore Time Point: **2019-12-23 17:47:00**

Step 3

To restore files and directories, drag and drop the selection to Restore Selection Area.

Backup Content

Path: /

Name	Size	Date
home/	20 B	2019-12-23 16:38:40
media/	6 B	2018-04-11 06:59:55
mnt/	6 B	2018-04-11 06:59:55
opt/	6 B	2018-04-11 06:59:55
proc/	6 B	2019-12-23 16:29:39
root/	114 B	2019-12-23 16:40:04
run/	6 B	2019-12-23 16:29:40
srv/	6 B	2018-04-11 06:59:55
sys/	6 B	2019-12-23 16:29:40
tmp/	211 B	2019-12-23 17:29:41
usr/	155 B	2019-12-23 16:30:01
var/	267 B	2019-12-23 17:29:07
bin	7 B	2019-12-23 16:30:01

Limit: 0 - 500 Search in elements...

Version Browser

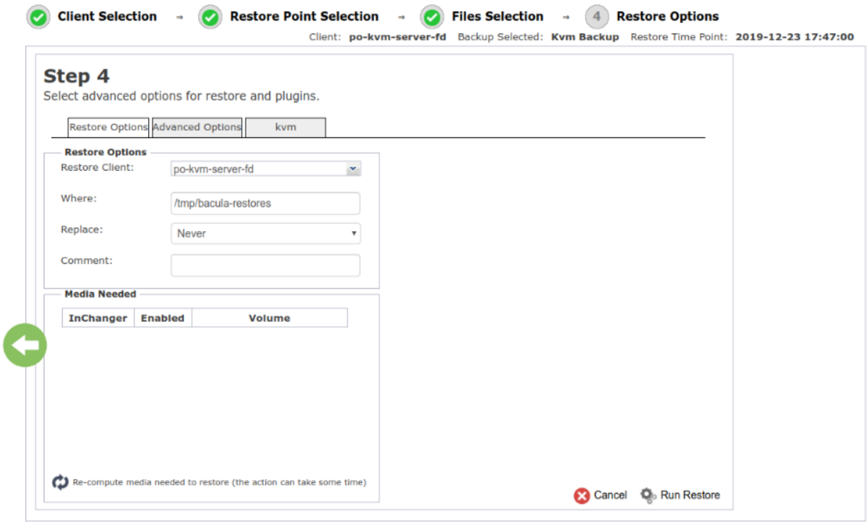
Here it is possible to choose between all versions of the current selected file, drag and drop selected version to the Restore Selection Area.

InChanger	Volume	Size	Date	Checksum

Restore Selection Area

It is possible to drop files and directories here from the Backup Content list or from the Version Browser list above.

Name	Jobid	Size	Date	Action
/etc/	15	8.0 KiB	2019-12-23 17:29:13	
/usr/	15	155 B	2019-12-23 16:30:01	



Client: **po-kvm-server-fd** Backup Selected: **Kvm Backup** Restore Time Point: **2019-12-23 17:47:00**

Step 4

Select advanced options for restore and plugins.

Restore Options

Restore Client: po-kvm-server-fd

Where: /tmp/bacula-restores

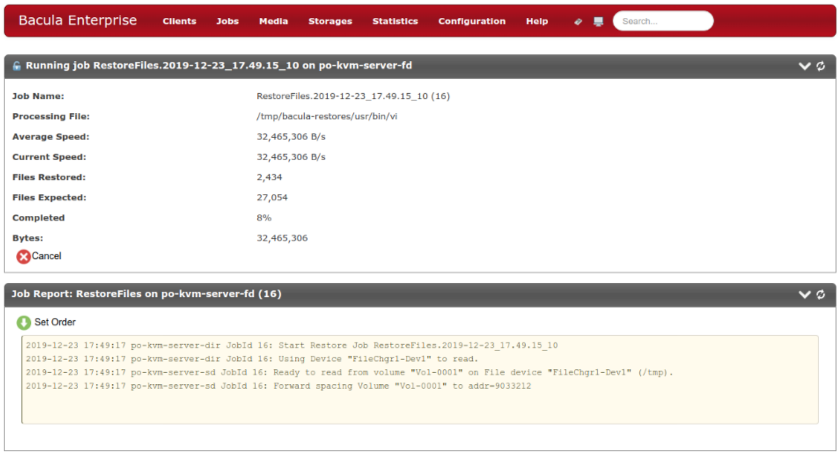
Replace: Never

Comment:

Media Needed

InChanger	Enabled	Volume

Re-compute media needed to restore (the action can take some time)



Running job RestoreFiles.2019-12-23_17.49.15_10 on po-kvm-server-fd

Job Name:	RestoreFiles.2019-12-23_17.49.15_10 (16)
Processing File:	/tmp/bacula-restores/usr/bin/vi
Average Speed:	32,465,306 B/s
Current Speed:	32,465,306 B/s
Files Restored:	2,434
Files Expected:	27,054
Completed	8%
Bytes:	32,465,306

Job Report: RestoreFiles on po-kvm-server-fd (16)

```
2019-12-23 17:49:17 po-kvm-server-dir JobId 16: Start Restore Job RestoreFiles.2019-12-23_17.49.15_10
2019-12-23 17:49:17 po-kvm-server-dir JobId 16: Using Device "FileChgr1-Dev1" to read.
2019-12-23 17:49:17 po-kvm-server-sd JobId 16: Ready to read from volume "Vol-0001" on File device "FileChgr1-Dev1" (/tmp).
2019-12-23 17:49:17 po-kvm-server-sd JobId 16: Forward spacing Volume "Vol-0001" to addr=9033212
```

See also:

- Go back to *KVM Backup*

Go back to *the main Operations page*.

Go back to *the main KVM Plugin page*.

See also:

- Go back to *KVM Scope*
- Go back to *KVM Features*
- Go back to *KVM Installation*
- Go back to *KVM Configuration*
- Go to *KVM Limitations*

Go back to *the main KVM Plugin page*.

Go back to the main Dedicated Backup Solution page.

6 Limitations

- When performing backup at the KVM domain controller (hypervisor) level, some Bacula features such as client RunScripts will not execute commands at the guest level. Dumping a MySQL database on a guest VM before the backup will require the use of custom scripts, using SSH for example or simply make us of the Bacula Enterprise MySQL Plugin. The same can be applied to any application running on the VM with any Bacula Enterprise plugin.
- Virtualization tools (qemu-ga) must be installed on the guest, and you may have to configure the guest to quiesce the file system properly during the snapshot phase.
- Virtual machines cannot be restored in a bare metal recovery way using the KVM Plugin.
- CD ROMS *must* be disconnected from VMs before attempting backup with the KVM Plugin. Else, the system might report the error "guestmount: multi-boot operating systems are not supported".
- The KVM Plugin uses snapshots while backing up guest VMs. During a snapshot, blocks modified by the guest VM need to be copied in temporary space, the space required depends on the guest disk activity. By default the space is allocated under `/var/tmp`.

- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

See also:

- Go back to *KVM Scope*
- Go back to *KVM Features*
- Go back to *KVM Installation*
- Go back to *KVM Configuration*
- Go back to *KVM Operations*

Go back to *the main KVM Plugin page*.

Go back to the main Dedicated Backup Solution page.

Go back to the main Dedicated Backup Solutions page.