

100% data integrity with ArchivistaVM backups

Egg, 6th March 2014: *the data storage (backup) options available with ArchivistaVM are described in this blog entry. The following options are available with both the free of charge ArchivistaBox Mini and the fee-based versions. The start up carried out conceptually and includes information on why the established backup methods currently used for virtualisation are beset with risks in serious cases and why data backup using ArchivistaVM is a far better solution. Step-by-step instructions are then given, so that data backup (including the retrieval of guests) can be carried out using ArchivistaVM.*



There is no secure data backup concept currently available for virtualisation!

The author is continually astonished at how naively experienced IT managers tackle the subject of data backup - especially with regard to virtualisation. **In virtualisation, so-called “hot-swap backups” (other names such as “snapshots” are also commonly used) are almost always implemented. “Hot-swap” means that, prior to backing up, the machine is put into the static condition, so that backing up (without guest shutdown) can then be carried out.**

In order that guests can be retrieved “without problems” at some later point, the main memory (RAM) and the hard drive files are normally backed up in such a way that the guests are temporarily “frozen in”. An image of the main memory (RAM) is then created and backed up together with the contents of the hard drive file(s). **During restore, the backed-up content of the main memory (RAM) and the hard drive files are shifted back to the time of the original backup, i.e. the guests are back-dated to the point in the operation when the hot-swap data backup was implemented.**

This ensures that the exact conditions that existed at the time of the data backup are recreated. **However, such backups cannot be regarded as static, because during hot-swap operations, no checks are carried out as to whether the data actually exists in a static condition.** In the worst case scenario, the **backup is in a state where the operating system itself has stalled. Alternatively, the backed-up files in the database server are corrupted;** in the best case scenario, these files can be retrieved, though retrieval is usually extremely time-consuming.

For database applications (which are the rule in the server environment), plugins are often installed in the guests, so that the databases are briefly paused

prior to the creation of the memory image (for example); this, however, requires plug-ins to be available and also that the corresponding versions, the virtualisation software and the guest software are compatible. So, what can go wrong? Very simply, the virtualised guests can only be backed up if the virtualisation software provider makes the operating system plug-ins available. Or to put it another way, if there is no plug-in being used for the operating system, no accurate data backups can be carried out. **Ultimately, this is sold as 'certified for XYZ', but it should be labelled 'limited to XYZ'.**

The ArchivistaVM concept

Things can be a lot simpler than the process described above. The host (virtualisation software) and the guest (virtualised operating system) operate autonomously, the **guest is briefly shut down during data backup, data backup is executed and the guest is rebooted.** If the guests are redundant on the hard drive, the time during which the guest is unavailable can be extremely short. Guest shutdown, 2nd hard drive unmounted, guest booted, backup executed, 2nd hard drive re-mounted.

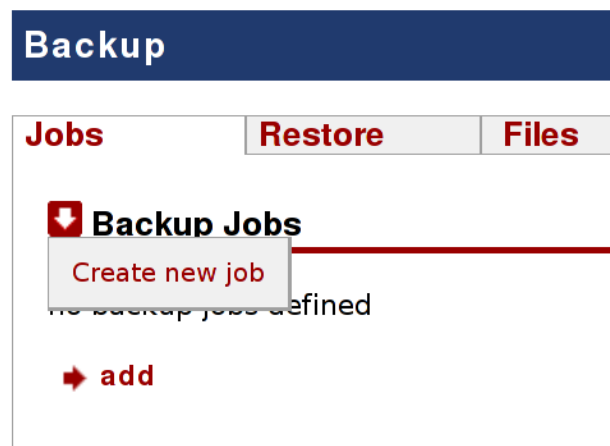
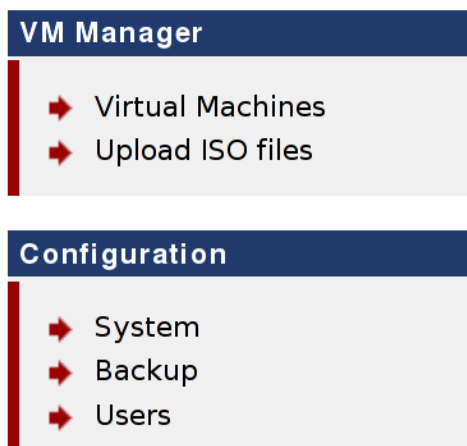
If no guest shutdown is carried out (in the Linux environment this is generally frowned upon, long uptimes are the pride of any administrator), the relevant scripts in ArchivistaVM can be started outside the guest before and after data backup, which thereby ensures that the data in the guest is in a static condition. But if we're being honest, even Linux operating systems are nowadays much more readily updated than they were previously, and here too it is becoming increasingly desirable to backup the entire guest, so as not to be in the position of having the wrong version of the database server when retrieving on some future "Day X".

The concept used by ArchivistaVM is sadly not used for virtualisation solutions, **although instances are 100% static and autonomous only in this concept.** The concept needs no additional data backup software, as many instances as required can be backed up retrospectively and there is no guest-dependency - in a nutshell, these are standard compliant hard drive files which, if necessary, can be booted or opened offline (**more on that to follow shortly**).

Note: Just to make sure that we all completely understand each other, ArchivistaVM can, of course, implement trouble-free hot-swap backups in the qcow2 format, but these backups contain precisely the same defects as the other solutions, so we won't highlight these further!

Backing up and retrieving guests with ArchivistaVM

Of course, guests (instances) can only be backed up if they have already been set up; if nothing has been set up, then there is nothing to back up. If this is the case, the data can be backed up in ArchivistaVM to the **left of the "backup" menu item and in "jobs"** by going down and **clicking on the "create new job" arrow**.



The data backup options can now be established. Note the “**Destination**” and “**External device**” fields. Provided no external device (hard drive) is present (which is the case in the following first attempts), the data is backed up into this directory on the internal hard drive.

Note: *If the data needs to be saved to an external hard drive at some later point, then naturally, in the context of these instructions, it would be advisable to first try out the simpler variant (no external drive necessary).*

The “**Keep old backups (1-x)**” field deserves special mention at this point. This allows specification of the number of backups before the last (oldest) backup is overwritten.

Backup

Jobs
Restore
Files

Create new job

Cluster Node: avbox179 (192.168.0.179)

Send email to:

Destination: /var/lib/vz/dump

Include all VMs:

External device:

Keep old backups (1-x): 5

Day of Week:

Mon	Tue	Wed	Thu	Fri	Sat	Sun
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Script before backup:

Start time: 02 : 00

Script after backup:

Mode: stop

VMID	Status	Name	VM Type	Disk size
<input checked="" type="checkbox"/> 101	stopped	test	qemu	0.00 GB

➔ save

Note: In order to **back up the data onto an external hard drive, the drive identifier must first be entered** (e.g. /dev/sdb1, /dev/sdc1) under “**external device**”. In this case, the target directory merely serves as the mounting point for the external device; the guests themselves are not saved in this target directory, but directly onto the external device. **The external drive identifier can be called using the “restore” tab.**

Note II: In contrast to earlier versions of the ArchivistaVM, **NTFS-formatted drives can be used** with the current version. However, it should be noted that when **using NTFS-formatted drives, no restore-on-the-fly is possible.**

Backup

Jobs | **Restore** | **Files**

Backup Jobs

Node	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Start time	Destination	Selection
avbox179	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	02:00	/var/lib/vz/dump	101

[Edit](#)
[Delete](#)
[Start now](#)

Data backup can now be set to the desired time (in this case 02:00 am). The job should, of course, be tested immediately. This is done by going down, **clicking on the arrow for the corresponding job and then selecting “Start now”**. This triggers a prompt, asking whether data backup should be started. Please confirm.

Backup

Jobs | **Restore** | **Files**

Confirm

Confirm backup?

The data backup will now be executed. The log files can be monitored under the “data backup” menu item.

Backup

Jobs | **Restore** | **Files**

Backup Jobs

Node	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Start time	Destination	Selection
avbox179	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	02:00	/var/lib/vz/dump	101

➔ add

```

Apr 08 20:42:05: Finished Backup of VM 101 (00:00:45)
Apr 08 20:42:05: file size 166M
Apr 08 20:41:20: now save file /var/lib/vz/images/101/vm-101-disk.raw
Apr 08 20:41:20: now save file /var/lib/vz/images/101/101.conf
Apr 08 20:41:20: keep old backups: 30
Apr 08 20:41:20: creating archive in '/var/lib/vz/dump/101'
Apr 08 20:41:20: Starting Backup of VM 101
Apr 08 20:41:19: Starting new backup job - vzdump --quiet --stop --dumpdir /var/lib/vz/dump --keepbackup 51 101

```

Once the operation is complete, the “finished backup” message is displayed.

Retrieving backed-up data

In order to retrieve previously backed-up data, use the “restore” tab next to “jobs” in the “data backup” sub-menu. This allows viewing of the existing backups.

Backup

Jobs

Restore

Files

External device:

From path:

From ID:

Version:

New ID:

➔ restore

ID: 101, Version: 1 (2014-04-08), name: test

In order to retrieve a backup, its ID, the correct version number and a new ID for the restored data must be entered. In our example, ID 101 is to be retrieved using Version 1 and restored to ID 201.

Backup

Jobs

Restore

Files

External device:

From path:

From ID:

Version:

New ID:

➔ restore

ID: 101, Version: 1 (2014-04-08), name: test

The operation can now be executed using the “**Restore**” tab. Here too, status reports are available under the “jobs” tab. Once the “**restore successful**” message is displayed, the operation is complete.

Backup

Jobs
Restore
Files

+ **Backup Jobs**

Node	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Start time	Destination	Selection
↓ avbox179	☑	☑	☑	☑	☑	☑	☑	02:00	/var/lib/vz/dump	101

→ add

```

Apr 08 20:50:07: restore successful
Apr 08 20:50:07: restore configuration to '/etc/qemu-server/201.conf'
Apr 08 20:49:28: space needed: 179436, available: 31667232
Apr 08 20:49:28: restore qemu-server image 101 using ID 201
Apr 08 20:42:05: Finished Backup of VM 101 (00:00:45)
Apr 08 20:42:05: file size 166M
Apr 08 20:41:20: now save file /var/lib/vz/images/101/vm-101-disk.raw
Apr 08 20:41:20: now save file /var/lib/vz/images/101/101.conf
Apr 08 20:41:20: keep old backups: 30
Apr 08 20:41:20: creating archive in '/var/lib/vz/dump/101'
Apr 08 20:41:20: Starting Backup of VM 101
Apr 08 20:41:19: Starting new backup job - vzdump --quiet --stop --dumpdir /var/lib/vz/dump --keepbackup 51 101

```

Note: For large guests (instances), this operation can take several hours. Depending on the hard drive and interface used (USB2/USB3), **average retrieval speeds of 50 - 200 MB per second can be expected.**

Note II: Instance retrieval cannot be started unaltered whilst another instance is in progress, since this would mean the same Mac- and IP addresses being used twice. The Mac-address can be altered using the “hardware” tab, by removing the network card and then replacing it with a new Mac-address. The guest can then be started and the IP-address(es) changed. Only then can the two instances be started (if, of course, this is actually necessary).

Backing up onto external devices (hard drives)

In order to **backup onto an external hard drive**, the **correct identifier (e.g.: /dev/sdb1, /dev/sdc1) for the job definition must be entered.** The simplest way is to **call up the identifier using the “restore” tab in the connected (not the mounted) drive.** Provided the drive is connected, the identifier will be visible under **“External device”** (in our example: /dev/sdb1).

Backup

Jobs	Restore	Files
External device:	<input type="text" value="/dev/sdb1"/>	
From path:	<input type="text" value="/var/lib/vz/dump"/>	
From ID:	<input type="text"/>	
Version:	<input type="text" value="1"/>	
New ID:	<input type="text"/>	
➔ restore		

A new backup can now be created under **“Jobs”**, by entering the identifier under **“External device”**.

Jobs	Restore	Files																		
Create new job																				
Cluster Node:	<input type="text" value="avbox179 (192.168.0.179)"/>	Send email to: <input type="text"/>																		
Destination:	<input type="text" value="/var/lib/vz/dump"/>	Include all VMs: <input type="checkbox"/>																		
External device:	<input type="text" value="/dev/sdb1"/>	Keep old backups (1-x): <input type="text" value="2"/>																		
Day of Week:	<table border="0"> <tr> <td>Mon</td><td>Tue</td><td>Wed</td><td>Thu</td><td>Fri</td><td>Sat</td><td>Sun</td> </tr> <tr> <td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td> </tr> </table>	Mon	Tue	Wed	Thu	Fri	Sat	Sun	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Script before backup: <input type="text"/>				
Mon	Tue	Wed	Thu	Fri	Sat	Sun														
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>														
Start time:	<input type="text" value="02"/> : <input type="text" value="00"/>	Script after backup: <input type="text"/>																		
Mode:	<input type="text" value="stop"/>																			
<table border="1"> <thead> <tr> <th></th> <th>VMID</th> <th>Status</th> <th>Name</th> <th>VM Type</th> <th>Disk size</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>101</td> <td>stopped</td> <td>test</td> <td>qemu</td> <td>0.00 GB</td> </tr> <tr> <td><input checked="" type="checkbox"/></td> <td>201</td> <td>stopped</td> <td>test</td> <td>qemu</td> <td>0.00 GB</td> </tr> </tbody> </table>				VMID	Status	Name	VM Type	Disk size	<input type="checkbox"/>	101	stopped	test	qemu	0.00 GB	<input checked="" type="checkbox"/>	201	stopped	test	qemu	0.00 GB
	VMID	Status	Name	VM Type	Disk size															
<input type="checkbox"/>	101	stopped	test	qemu	0.00 GB															
<input checked="" type="checkbox"/>	201	stopped	test	qemu	0.00 GB															
➔ save																				

Note: Provided that an entry exists under **“external device”**, the backup will immediately be executed onto an external hard drive. If the entry is inconsistent, the backup cannot be executed; this is then reported in the log file.

Note II: The two options **“stop”** and **“cluster”** are available under **“mode”**. The **“cluster”** option allows backups to be created in clustered servers (a minimum of two ArchivistaVM servers). An instance to be backed up on the first node ends up on the last node, all remaining instances end up on the previous node. Here it must be ensured that the hard drive is connected to the correct node. For example: If an instance to be backed up onto the first node of a double cluster ends up on the hard drive of the second node; this suggests that the hard drive connection to the first node is non-functional.



Facebook



Twitter