



Hyper-V WinAPI Plugin

Bacula Systems Documentation

Contents

1	Hyper-V Plugins to Cover All Backup Needs	2
1.1	Features summary	3
1.2	Important notes	3
1.3	Supported platforms	4
1.4	Installation	4
1.5	Important Considerations regarding Credentials Settings	6
1.6	Job Configuration	12
1.7	Examples	13
1.8	Backup	14
1.9	Restore	15
2	Best Practices	19
2.1	Failover Cluster	20
2.2	Recovery	20

Contents

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

Important: The HyperV WinAPI plugin is the newest and most advanced plugin available for backing up Hyper-V environments. It offers the most comprehensive set of features, ensuring seamless integration and efficient backups. Due to its robust functionality and extensive capabilities, this plugin is now the recommended solution for all Hyper-V backup needs.

1 Hyper-V Plugins to Cover All Backup Needs

Microsoft has created various technologies for backing up Hyper-V virtual machines. Thus, multiple Bacula Plugins have been designed to maximize the benefits of each solution.

Hyper-V is equipped with a VSS writer on all compatible versions of Windows Server. This VSS writer enables developers to utilize the existing VSS infrastructure for backing up virtual machines to Bacula using the Bacula Enterprise VSS Plugins. This technology is supported by the original **Bacula Hyper-V plugin**. Although it doesn't allow incremental and differential backups and other more granular VM-based options, it can still cover any Hyper-V version. Therefore, it is highly recommended for small standalone Hyper-V servers (single node, no Failover Cluster) and older Hyper-V versions that do not offer Virtual Disk Service or Hyper-V WMI API.

The Virtual Disk Service (VDS) is a service provided by Microsoft Windows that handles query and configuration tasks upon request from end users, scripts, and applications. This service is compatible with Windows Server 2003, Windows Vista, and newer versions. The **Hyper-V Winapi Plugin** utilizes this technology to backup and restore virtual machines. It supports incremental and differential backups, making it the recommended solution for more intricate Hyper-V servers, such as Failover Clusters with multiple nodes, where local disk space is a critical resource. Depending on the relocation of virtual machines within the Cluster, it may be necessary to migrate the backed-up VMs across specific nodes.

Starting in Windows 8 and Windows Server 2012, Hyper-V supports backup via the Hyper-V WMI API. This feature enables individual Guest VMs to be backed up separately and incrementally, offering a more scalable solution compared to using VSS in the host. The Bacula Enterprise **Hyper-V WMI Plugin** uses this technology for backup and restore to/from Bacula. It backs up/restores VM in the recommended Microsoft format. By utilizing the Microsoft snapshot format from/to disk, it is essential to have sufficient disk space available for the process to proceed smoothly. In scenarios where disk space may be limited due to busy configurations, the Hyper-V Winapi Plugin can be utilized as an alternative solution.

Backups created using the **Hyper-V WMI Plugin**, **Hyper-V Winapi Plugin** and **Hyper-V Plugin** are not compatible with each other.

1.1 Features summary

- Backups are performed on virtual machines based on VMs, with a configuration that relies on Failover Cluster. After migrating to the local node, remote VMs can be backed up. The migration process to or from the local node is automated.
- The hosted VMs are backed up using Full, Differential, and Incremental backup methods, without the need for exporting local snapshots. This approach helps to avoid excessive disk space usage for backups.
- When restoring, the complete virtual machine images are directly placed in their original location, eliminating the need for additional disk space.

1.2 Important notes

- The **Hyper-V WinAPI Plugin** exclusively supports local VMs. Therefore, remote VMs are automatically **migrated** to the local node, backed up, and then migrated back to their original node. For more information, refer to the Backup section.
- To perform backups using the **Hyper-V WinAPI Plugin**, the **accurate** option must be enabled.
- It is highly recommended to enable **compression** when using the **Hyper-V WinAPI Plugin** for backups.
- Attempting multiple backups of the same VM in parallel is not advisable. Each backup will attempt to recover ongoing actions from the previous one, which can result in errors.
- Backups created with the **Hyper-V WinAPI Plugin** are not compatible with Virtual Full jobs. It is important not to combine these two backup strategies as you will not be able to properly restore jobs from Virtual Full backups.
- Similarly, backups created with the **Hyper-V WinAPI Plugin** are not compatible with backups created using the **Hyper-V WMI plugin** or **Bacula Hyper-V Plugin**.
- The **Hyper-V WinAPI Plugin** requires Hyper-V Virtual Machines version 6.2 or higher to correctly handle Differential and Incremental backups.
- 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.

1.3 Supported platforms

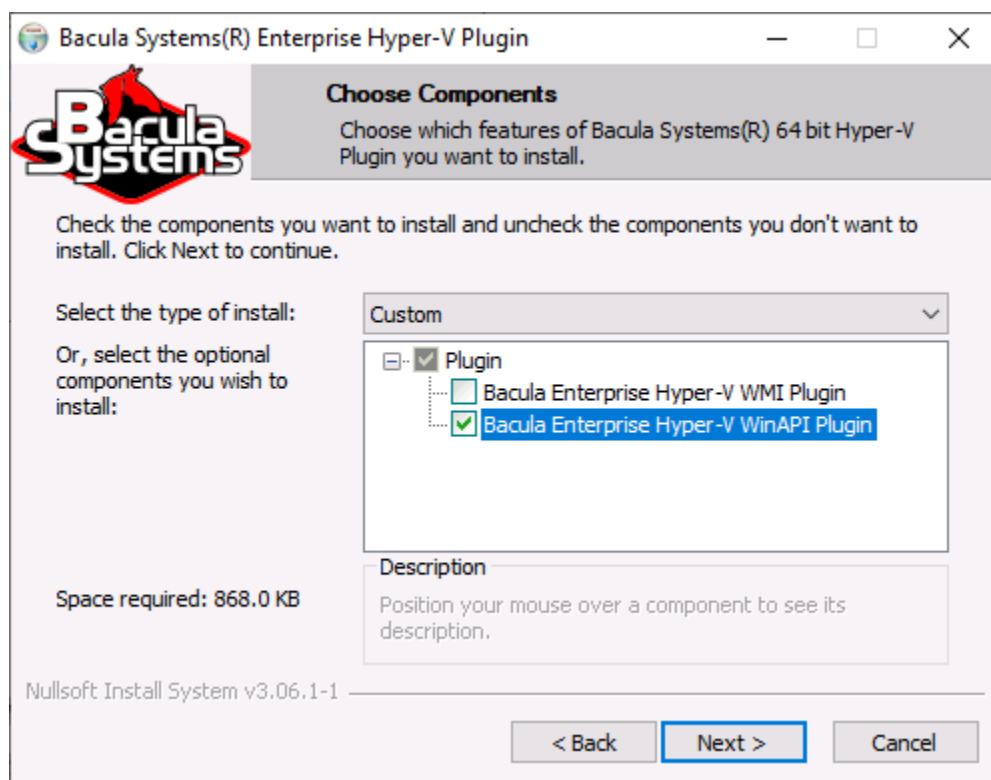
This documentation presents solutions for **Bacula Enterprise** 18.0 and higher, and is not applicable to prior versions of Bacula.

This Plugin supports Windows 8, Windows Server 2012 and later (Hyper-V Powershell module is required).

1.4 Installation

The Bacula File Daemon and the **Hyper-V WinAPI Plugin** need to be installed on the Hyper-V host server. The **Hyper-V WinAPI Plugin** Windows installer is the same as the **Hyper-V WMI Plugin** installer.

You can choose to install one or the other from the Plugin tree.



It will deploy required components within the Bacula File Daemon plugins directory.

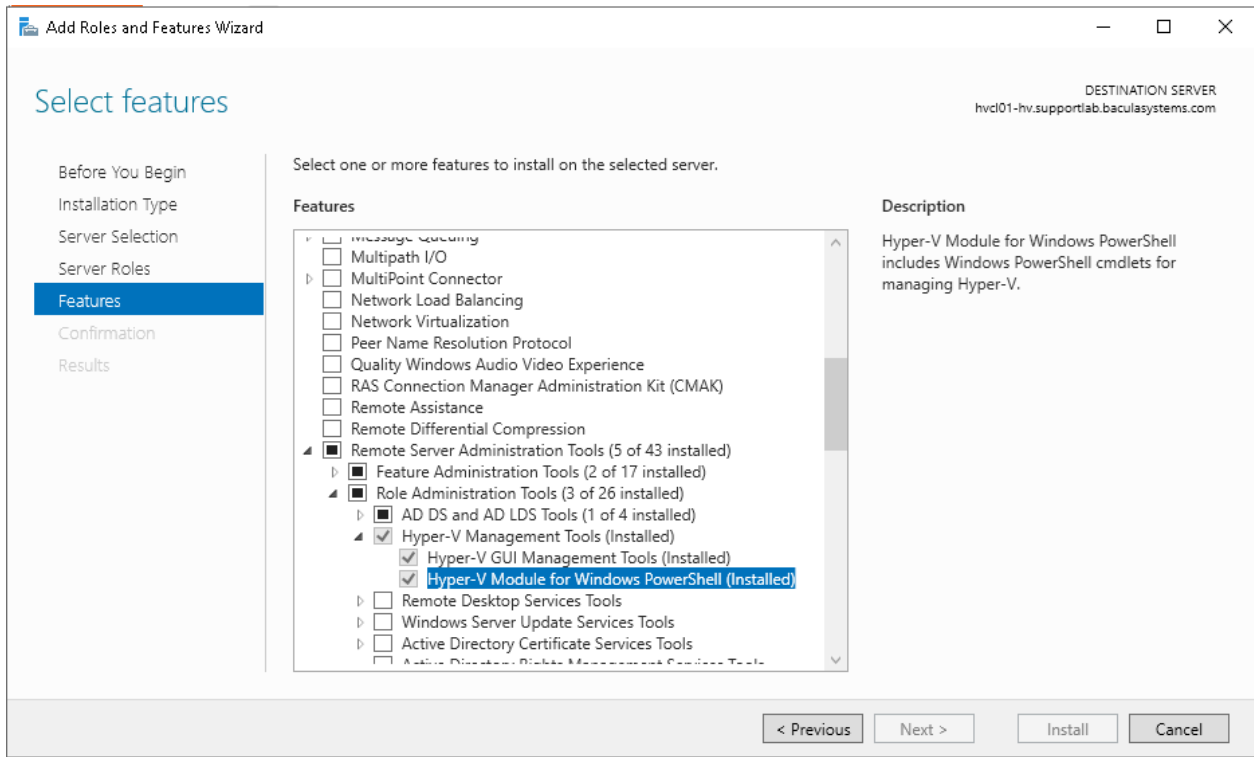
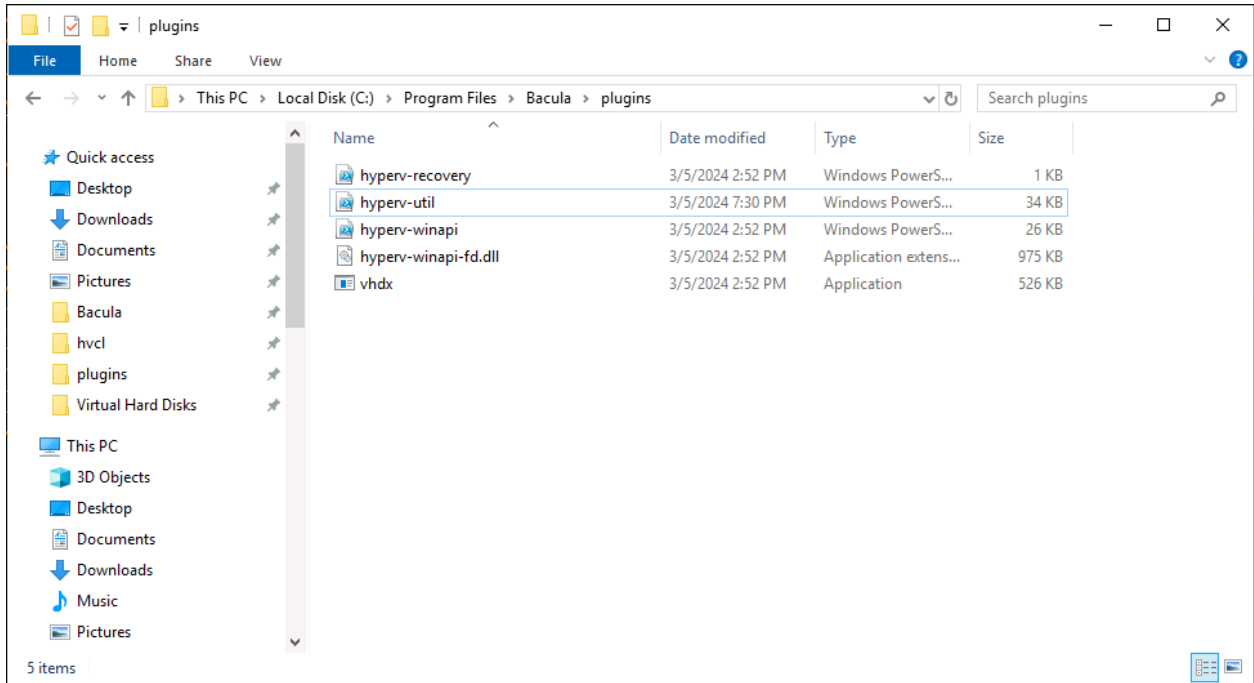
To configure the Bacula File Daemon, refer to the general Bacula installation documentation.

On the server side, the *Hyper-V PowerShell Module* needs to be enabled. On Windows Server or Hyper-V server 2012, 2016 and 2019, use Server Manager to install it. It should be located under Remote Server Administration Tools -> Role Administration Tools -> Hyper-V Management Tools and check Hyper-V Module for Windows PowerShell.

Verify the correct installation of the FD and the **Hyper-V Plugin** by running status client from bconsole or from BWeb.

```
*status client=w2019-hv01-fd
Connecting to Client w2019-hv01-fd at 172.22.22.50:9102
w2019-hv01-fd Version: 16.8.0 (06 April 2021) VSS Linux Cross-compile Win64
Daemon started 19-Jun-21 16:31. Jobs: run=23 running=2.
Microsoft Windows 2012 Standard Edition (build 9200), 64-bit
Priv 0x73f
```

(continues on next page)



(continued from previous page)

```
Memory: WorkingSetSize: 34,168,832 QuotaPagedPoolUsage: 183,768 QuotaNonPagedPoolUsage: 17,368 PagefileUsage: 43,687,936
APIs=OPT,ATP,LPV,CFA,CFW,
WUL,WMKD,GFAA,GFAW,GFAEA,GFAEW,SFAA,SFAW,BR,BW,SPSP,
WC2MB,MB2WC,FFFA,FFFW,FNFA,FNFW,SCDA,SCDW,
GCDA,GCDW,GVPNW,GVNFVMPW,LZO,EFS
Heap: heap=34,168,832 smbytes=39,489,074 max_bytes=69,259,353 bufs=395 max_bufs=396
Sizes: boffset_t=8 size_t=8 debug=10 trace=1 mode=0,2010 bwlimit=0kB/s
Crypto: fips=no crypto=OpenSSL
APIs: !GPFS
Plugin: alldrives-fd.dll(1.2) hyperv-winapi-fd.dll(0.1) winbmr-fd.dll(3.1.0)
```

Verify that **hyperv-winapi-fd.dll** is in the “Plugin” line (last line in the above example output).

FIXME: NOTE FOR REVIEW Not sure about the behavior in the case of a Failover Cluster configuration, the Bacula file daemon and the **Hyper-V Winapi plugin** need to be installed on only one node or on several nodes to avoid automatic migrations (need some USER CASES here).

1.5 Important Considerations regarding Credentials Settings

Important: In order to access and backup the Hyper-V server, the delegation of the User credentials must be enabled and the Bacula File Daemon must be logged as an authorized user within the Hyper-V server.

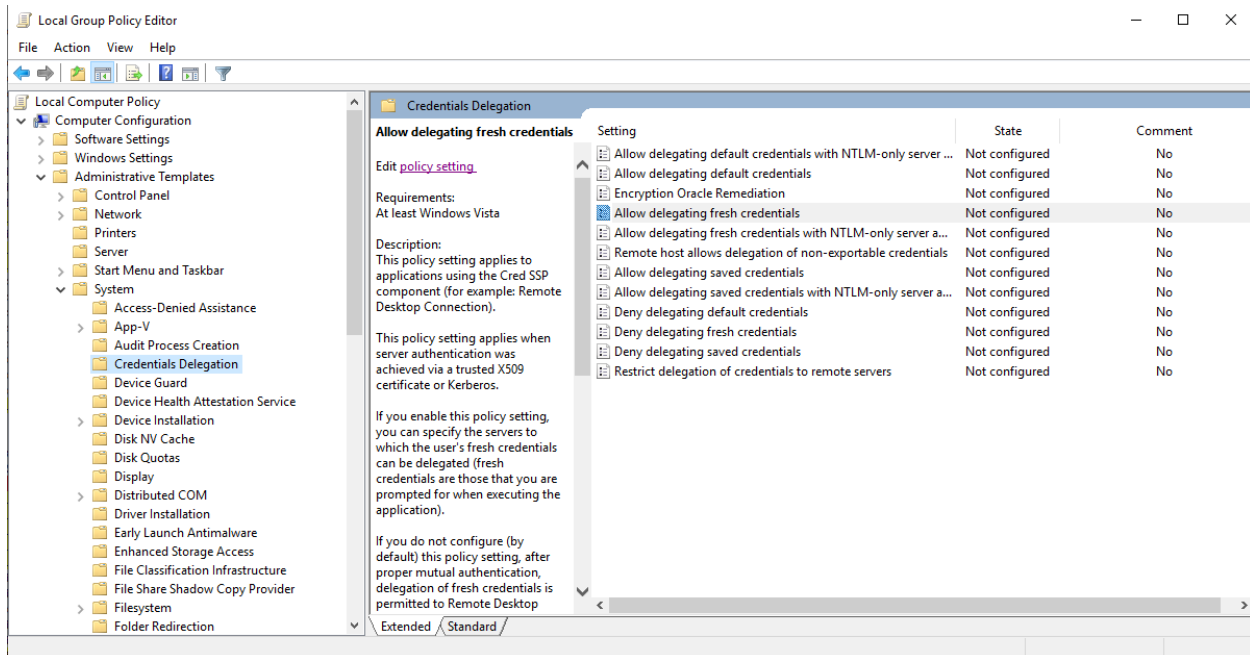
Enable Delegation of User Credentials on Hyper-V Server

- Run `gpedit.msc` (normally in `C:\Windows\System32`) on the Hyper-V server and look at the following policy: Computer Configuration -> Administrative Templates -> System -> Credentials Delegation -> Allow Delegating Fresh Credentials.
- Verify that it is enabled and configured with the WSMAN SPN appropriate for the target computer.

For example, for a target computer name “myserver.domain.com”, the SPN can be one of the following: `WSMAN//myserver.domain.com` or `WSMAN//*.domain.com`. Introduce it in the “Add servers to the list” “Show” dialog box.

- Finally run a powershell console on the Hyper-V server (normally in `C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe`) and enter the following commands:

```
Enable-WSManCredSSP -Role Server -Force
Enable-WSManCredSSP -Role "Client" -DelegateComputer myserver.domain.com -Force
```



Impersonation of Hyper-V WinAPI Plugin

The impersonation of the **Hyper-V WinAPI plugin** can be achieved in different ways.

- Specify the user name and password locally on the hyper-v node. This is the **recommended method**. In a `bacula-hyperv.pwd` file, located by the `bacula-fd.conf` config file (typically `C:\Program Files\Bacula`), `bacula-hyperv.pwd` contains the user name followed by the user password, separated by a colon.

```
name@domain.com:mypassword
```

or

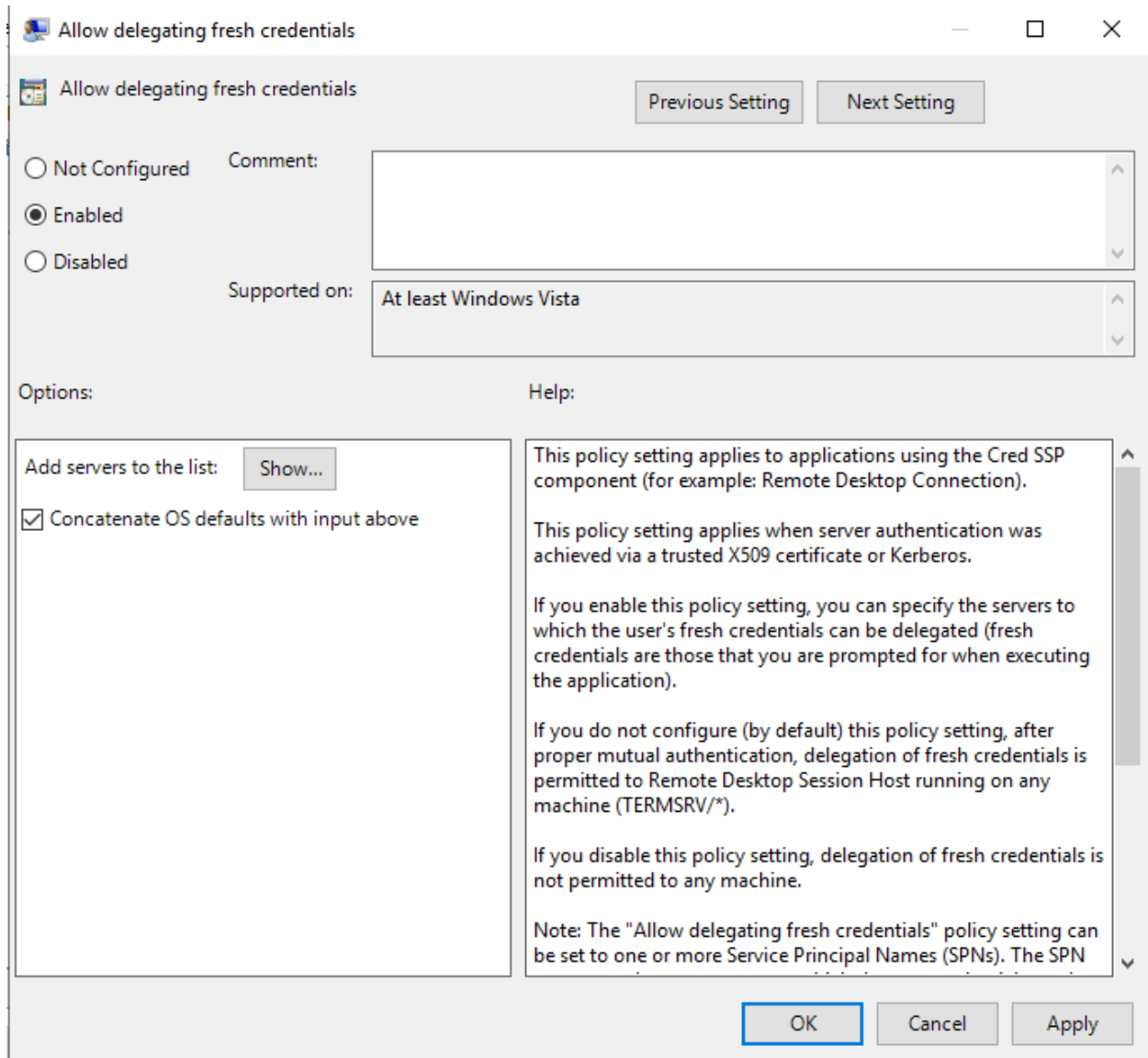
```
DOMAIN\name:mypassword
```

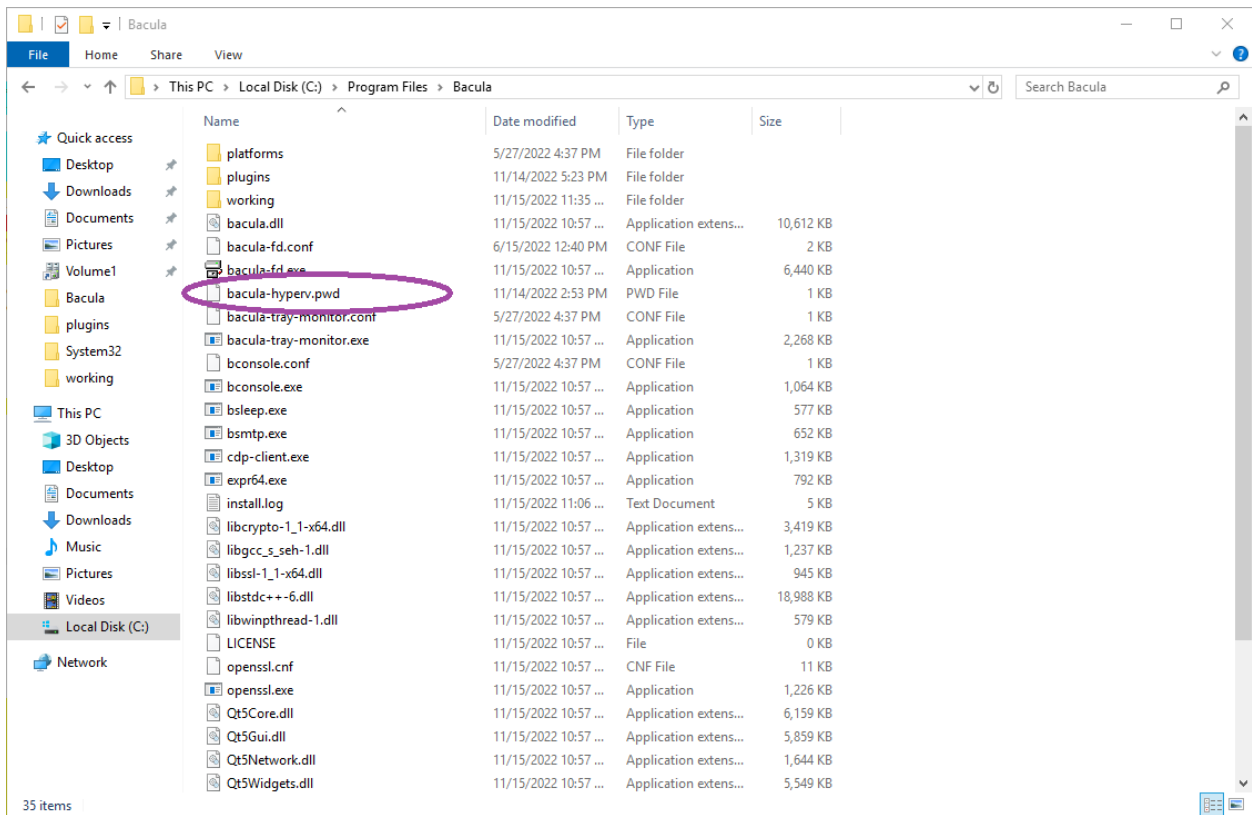
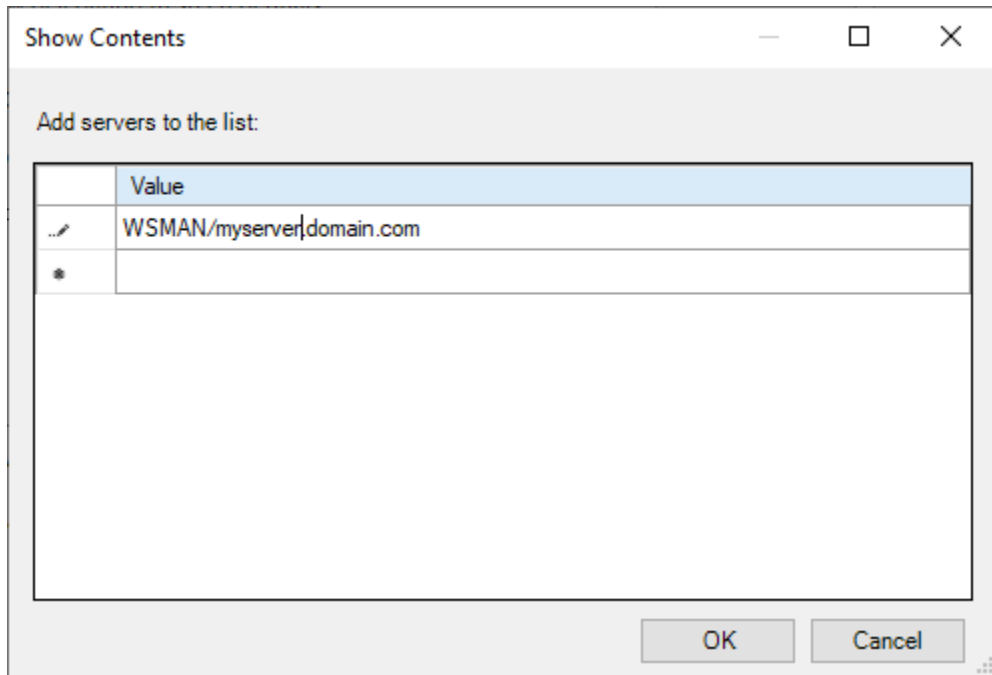
- Impersonate the **Hyper-V WinAPI Plugin** by passing user and password, as plugin options. See Job configuration `user_name` and `user_password` options.
- Manually change the Bacula File Daemon default login account:

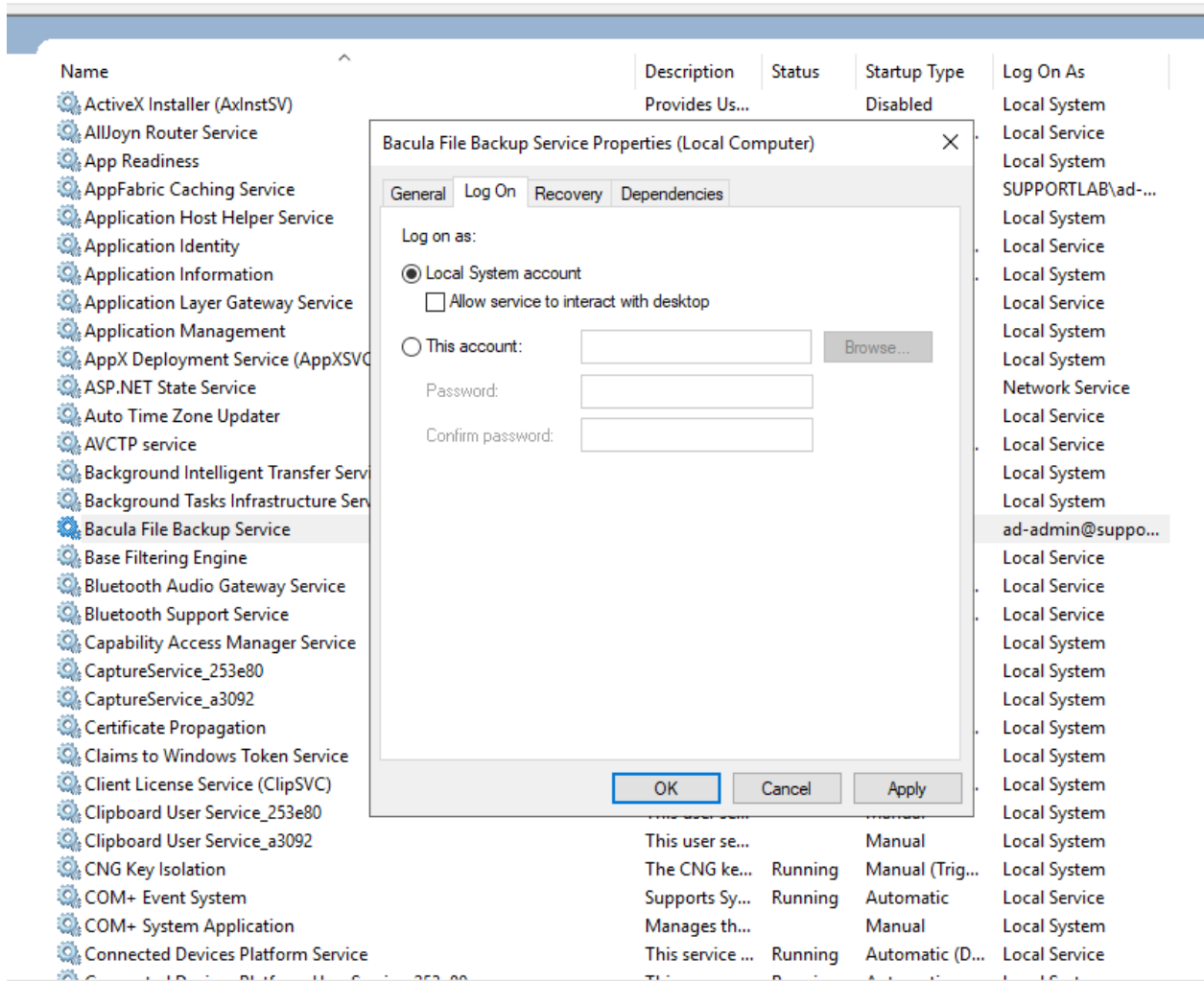
Access the Hyper-V server using administrative credentials. Go to the Windows Start menu, type in "Services", and press Enter to display a list of all installed services. Locate the Bacula File Backup Service, right-click on it, and select Properties. Then, navigate to the Log On tab. The settings should appear as follows:

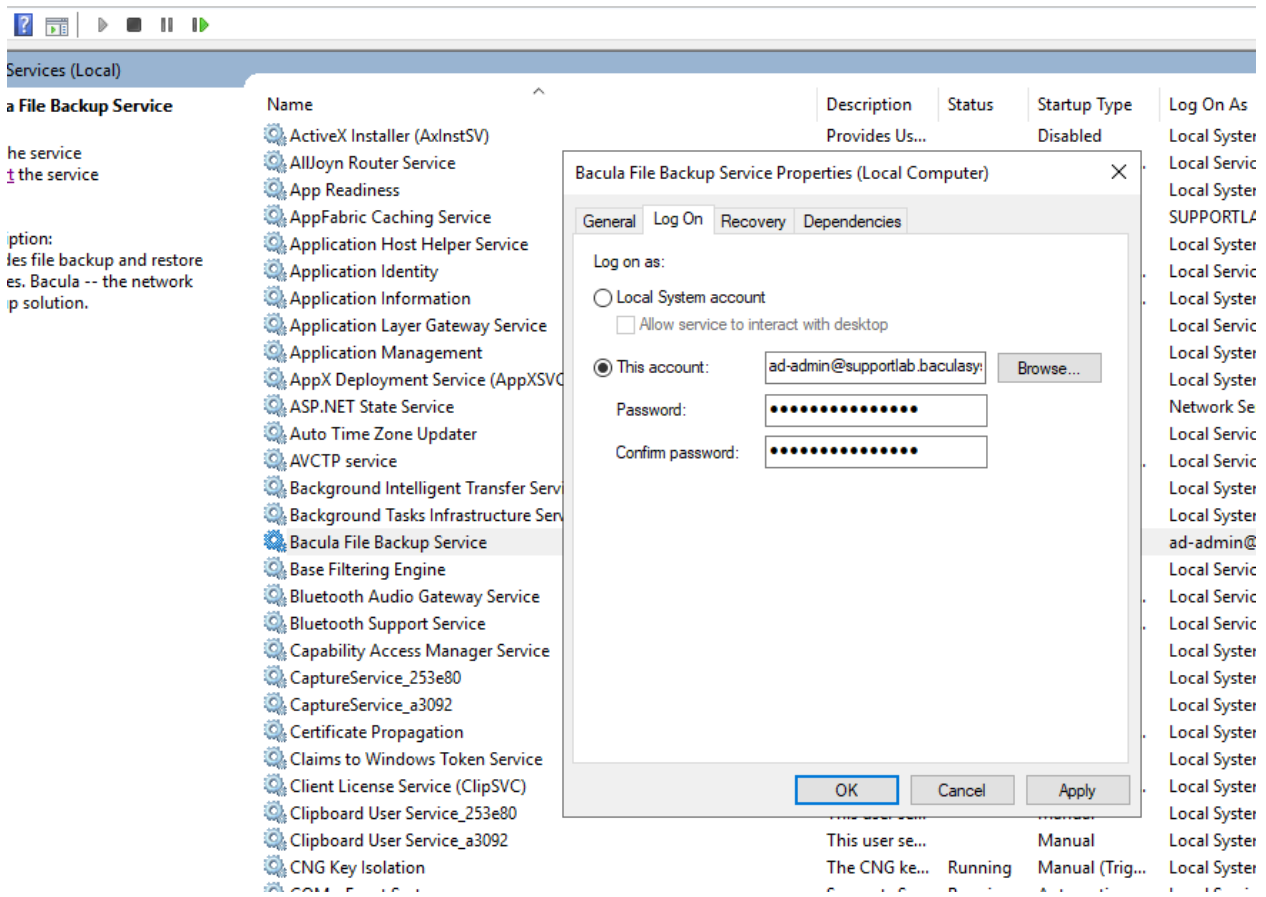
Toggle the selection from "Local System account" to "This account". Enter the credentials of a Hyper-V administrator (either read only or read-write). Click OK.

Click on the Bacula File Backup Service entry once more with the right mouse button, then select "Restart" to ensure that the changes take effect.









1.6 Job Configuration

Once the Bacula File Daemon and the **Hyper-V WinAPI Plugin** are correctly installed and configured, setting a backup job up is as simple as adding the job and the fileset within the Bacula Director configuration file.

Important: The `Enable VSS` parameter must be set to `no` in the FileSet (see examples below).

The following plugin options are supported:

Name	Status	Default	Description
<code>include</code>	Optional	Include all (*)	a Unix shell-style wildcards pattern for including VMs by name
<code>exclude</code>	Optional	Exclude none	a Unix shell-style wildcards pattern for excluding VMs by name
<code>tmp_dir</code>	Optional	Bacula-repo	changes the Bacula working repository folder. Make sure there's enough space on this location to create VM's snapshots and exports. Default is a <code>Bacula-repo</code> folder in the VHD location.
<code>pre_backup_action</code>	Optional	None	action on the VMs before backup takes place. Can be None, Stop, Save. - None is noop. - Stop stops the VM before backup (useful when VM doesn't support VSS or kernel freeze to maintain consistent backups). - Save saves the VM before stopping it.
<code>post_backup_action</code>	Optional	None	action on the VMs after backup is completed. Can be None, Restart, ForceRestart. - None is noop. - Restart restarts the VM if it was stopped or saved pre-backup. - ForceRestart restarts the VM unconditionally.
<code>consistency_level</code>	Optional	Application	overwrites the consistency level. Can be Application Consistent or Crash Consistent. Application is the recommended value but some VMs might not support it.
<code>allow_pre_snapshot</code>	Optional	Disabled	when enabled, allows retry with <code>pre_backup_action</code> set to Save and <code>post_backup_action</code> set to Restart, if crash consistency retry backup has failed.
<code>cluster_mode</code>	Optional	Disabled	when enabled, the plugin will parse and allow migration of VM's that are not on the local node (for Failover Cluster configuration).
<code>abort_on_error</code>	Optional	Disabled	abort immediately the job if a serious error is found (b.e when no VM matches the <code>include</code> patterns). By default, a Job error is raised, but the job continues.
<code>wait_on_vm_migration</code>	Optional	Disabled	when the VM is in "migrating" state, the plugin will wait for the VM migration completion before processing with the backup if the option is enabled. Otherwise, it skips the VM.
<code>user_name</code>	Optional	None	the user name that will run the backup/restore operation. This is not the recommended method . The user name can be specified locally on the hyper-v node in a <code>bacula-hv.user</code> file located in the <code>fd plugins</code> folder.
<code>user_password</code>	Optional	None	the user password that will run the backup/restore operation. This is not the recommended method . The user password can be specified locally on the hyper-v node in a <code>bacula-hv.pwd</code> file located in the <code>fd plugins</code> folder.

1.7 Examples

Example 1: backup all vms using Bacula's default working directory

```
Job {
  Name = "Hyper-V-BackupAll"
  Type = Backup
  Client= w2019-hv01-fd
  FileSet="Simplest-Hyper-V-FileSet"
  Storage = File
  Messages = Standard
  Pool = Default
  Accurate = yes
}

FileSet {
  Name = "Simplest-Hyper-V-FileSet"
  Enable VSS = no
  Include {
    Options {
      signature=MD5
      compression=GZIP
    }
    Plugin = "hyperv-winapi:"
  }
}
```

Example 2: backup only «Linux- » prefixed vms using Bacula's default working directory

```
Job {
  Name = "Hyper-V-BackupOnlyLinux"
  Type = Backup
  Client= w2019-hv01-fd
  FileSet="Linux-Hyper-V-FileSet"
  Storage = File
  Messages = Standard
  Pool = Default
  Accurate = yes
}

FileSet {
  Name = "Linux-Hyper-V-FileSet"
  Enable VSS = no
  Include {
    Options {
      signature=MD5
      compression=GZIP
    }
    Plugin = "hyperv-winapi: include=Linux-*"
  }
}
```

Example 3: backup any VM having «Windows» in its name, using a custom working directory

```
Job {
  Name = "Hyper-V-BackupOnlyWindowsOnF"
  Type = Backup
  Client= w2019-hv01-fd
  FileSet="WindowsOnF-Hyper-V-FileSet"
  Storage = File
  Messages = Standard
  Pool = Default
  Accurate = yes
}
FileSet {
  Name = "WindowsOnF-Hyper-VFileSet"
  Enable VSS = no
  Include {
    Options {
      signature=MD5
      compression=GZIP
    }
    Plugin = "hyperv-winapi: include=\"*Windows*\" tmp_dir=\"F:/backup\""
  }
}
```

1.8 Backup

If you choose not to utilize the **cluster_mode** option in your filesets for a Failover Cluster configuration, your backups will be limited to the VMs on the node where the File Daemon is installed, as long as they match your include/exclude settings. However, you have the option to install a file daemon on each node. On the other hand, if you do specify **cluster_mode**, the **Hyper-V WinAPI Plugin** will identify all VMs that meet your include/exclude settings across the entire cluster, including remote nodes that are accessible. It will then attempt to migrate remote VMs locally before performing the backup. Once the backup is complete, the VM will be automatically migrated back to its original node.

The files backed up from the Hyper-V server will be visible in a **bconsole** or will have the prefix `/@HYPERV-WINAPI/`.

Typically, a VM backup data is organized as follows:

```
/@HYPERV-WINAPI/0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9-test1
/@HYPERV-WINAPI/0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9/IDE00.out
/@HYPERV-WINAPI/0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9/IDE00.bhd
/@HYPERV-WINAPI/0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9/IDE00.bmp
/@HYPERV-WINAPI/0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9/IDE01.out
/@HYPERV-WINAPI/0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9/IDE01.bhd
/@HYPERV-WINAPI/0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9/IDE01.bmp
/@HYPERV-WINAPI/0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9/test1.bvm
```

Where:

- 0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9 is the VM UID of the “test1”
- 0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9-test1 is a convenience empty file reminding us that the content of the VM named test1 is saved into folder /0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9/
- each drive is named after its controller type (IDE, SCSI), its controller number and its location in the controller. IDE00 meaning: IDE controller number 0, location 0.

- each .bhd file contains attributes information on the drive
- each .out file is the actual content of the drive
- each .bmp file gives information on how the content of the .out file maps onto the original drive and is needed for Single Item Restore
- the .bvm file backups information on the virtual machine itself

Incremental-Differential backups: the **Hyper-V WinAPI Plugin** will automatically follow the backup level strategy as scheduled in Bacula.

Consistency Level: A backup can fail when the option “Application Consistent” is required for a VM that doesn’t support it.

- If an Application Consistent backup fails, the **Hyper-V WinAPI Plugin** will change automatically the Consistency Level to “Crash Consistent” and retry.
- Only if `allow_pre_save` is enabled, when a “Crash Consistent” backup fails, the **Hyper-V WinAPI Plugin** will change the `pre_backup_action` to “Save” and `post_backup_action` to “Restart” and retry.
- Migration from a distant node, current backup checkpoint and `pre_backup_action` are stored into the corresponding VM notes in a proprietary format. They are removed when the backup is successful. If some interruption occurs during the backup, they can be recovered afterwards.
- If none of the above works, the backup fails with Error.
- Some reference points are maintained on Hyper-V side by Bacula, corresponding to different snapshots backed up. Bacula will prune those reference points when needed to keep at most 4.

1.9 Restore

To ensure a successful restore, it is advisable to either choose the entire fileset or restore all files associated with a particular drive, rather than cherry-picking individual files from the backup.

Restore parameters:

- **Where:** Can specify a path for VM restoration. If the content is not a path (does not contain slashes or backslashes), it’s considered to be the new VM restore name.
- **New Virtual Machine Name:** For renaming the restored VM.
- **Restore Path:** Specify the location where Snapshot files are restored.
- **Name used to process restore:** impersonation user name (see Impersonation of the **Hyper-V Winapi Plugin**).
- **Password used to process restore:** impersonation user password (see Impersonation of the **Hyper-V Winapi Plugin**).

Restore Options	Advanced Options	Hyperv-wmi
-----------------	------------------	------------

Restore Options

Restore Client:


Where:

Replace:

Comment:

Media Needed

InChanger	Enabled	Volume
Click "Re-compute the Media" button to display the list of media that will be used during the restore.		

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

Restore Options	Advanced Options	Hyperv-Winapi
-----------------	------------------	---------------

hyperv-winapi: include="Norbert-tiny" tmp_dir="C:/ClusterStorage/Volume1/backup"

New Virtual Machine name

Restore Path

Name used to process restore

Password used to process restore

VM Renaming:

If the 'New Virtual Machine Name' is specified, the restored VM(s) will be renamed accordingly. In case the 'Where' is not a path value, the 'Where' value will be used for renaming the restored VMs. If neither is set, the restored VM(s) will retain their original name(s), with the restore job name in parenthesis. For instance, a VM named 'tiny-vm' restored by Job 'RestoreFiles.2023-10-18_14.39.28_32' will be named 'tiny-vm (from RestoreFiles.2023-10-18_14.39.28_32)' by default.

Restore Path:

If the 'Restore Path' is specified, it will be utilized as the restore path for all backup files related to the drive. However, if the 'Restore Path' is empty and the 'Where' path is set with a value, then the 'Where' path will be used instead. In the event that neither of the aforementioned conditions are met, the default locations on the hyper-v host will be used, without creating a specific folder named after the Bacula job name. It is important to note that during the restore process, the restore files will not be moved, therefore the Restore Path will serve as the location for the restored VM. In order to facilitate multiple restorations, the files will be restored in a designated folder named after the Bacula job name.

Boot Order:

The original boot order is restored automatically for generation 1 and 2 VMs. For generation 2 VMs, UEFI boot devices will not be visible in the Firmware list before the new VM is booted once.

Examples:

- Defaults:
 - Where: Empty
 - New Virtual Machine Name: Empty
 - Restore Path: Empty

The restored VM(s) are (re)created in the local Hyper-V host default VMs location, with original name followed by something like (from *RestoreFiles.<date>_<time>*).

- Quick Rename:
 - Where: newVMName
 - New Virtual Machine Name: Empty
 - Restore Path: Empty

The restored VM(s) are (re)created in the local Hyper-V host default VMs location and renamed newVMName.

- Large Restore:
 - Where: Empty
 - New Virtual Machine Name: NewVMName
 - Restore Path: C:\LargeStorage\restore

The restored VM(s) are (re)created and renamed NewVMName. Virtual drive(s) and VM files are located into a folder named after the JobName in C:\LargeStorage\restore. Something like : C:\LargeStorage\restore\RestoreFiles.<date>_<time>.

A typical restore console output will look as follows:

```

2023-10-18 14:39:30 bp-vsir-bweb102-dir JobId 21365: Start Restore Job RestoreFiles.2023-
↳10-18_14.39.28_32
2023-10-18 14:39:30 bp-vsir-bweb102-dir JobId 21365: Restoring files from JobId(s) 21364
2023-10-18 14:39:30 bp-vsir-bweb102-dir JobId 21365: Connected to Storage "File1" at 10.
↳0.98.5:9103 with TLS
2023-10-18 14:39:30 bp-vsir-bweb102-dir JobId 21365: Using Device "FileChgr1-Dev1" to
↳read.
2023-10-18 14:39:30 bp-vsir-bweb102-dir JobId 21365: Connected to Client "hvcl02-winapi"
↳at 10.0.97.22:9102 with TLS
2023-10-18 14:39:03 hvcl02-winapi JobId 21365: Connected to Storage at 10.0.98.5:9103
↳with TLS
2023-10-18 14:39:30 bp-vsir-bweb102-sd JobId 21365: Ready to read from volume "Vol-3263"
↳on File device "FileChgr1-Dev1" (/bck/a1).
2023-10-18 14:39:03 hvcl02-winapi JobId 21365: StartRestoreJob: Restoring vm in c:\
↳Volume1\restore\RestoreFiles.2023-10-18_14.39.28_32\
2023-10-18 14:39:30 bp-vsir-bweb102-sd JobId 21365: Forward spacing Volume "Vol-3263" to
↳addr=3877321766
2023-10-18 14:39:47 bp-vsir-bweb102-sd JobId 21365: End of Volume "Vol-3263" at
↳addr=5368645668 on device "FileChgr1-Dev1" (/bck/a1).
2023-10-18 14:39:47 bp-vsir-bweb102-sd JobId 21365: Ready to read from volume "Vol-0396"
↳on File device "FileChgr1-Dev1" (/bck/a1).
2023-10-18 14:39:47 bp-vsir-bweb102-sd JobId 21365: Forward spacing Volume "Vol-0396" to
↳addr=271
2023-10-18 14:39:56 bp-vsir-bweb102-sd JobId 21365: End of Volume "Vol-0396" at
↳addr=1732770639 on device "FileChgr1-Dev1" (/bck/a1).
2023-10-18 14:39:56 bp-vsir-bweb102-sd JobId 21365: Elapsed time=00:00:26, Transfer
↳rate=123.8 M Bytes/second
2023-10-18 14:40:06 bp-vsir-bweb102-dir JobId 21365: Bacula Enterprise bp-vsir-bweb102-
↳dir 16.0.7 (11Jul23):
  Build OS:          x86_64-redhat-linux-gnu-bacula-enterprise redhat (Core)
  JobId:             21365
  Job:               RestoreFiles.2023-10-18_14.39.28_32
  Restore Client:    "hvcl02-winapi" 16.0.7 (05Oct23) Windows Server 2019 Standard
↳ServerStandard (build 17763), 64-bit,Cross-compile,Win64
  Where:             c:\Volume1\restore\
  Replace:           Never
  Start time:        18-Oct-2023 14:39:30
  End time:          18-Oct-2023 14:40:06
  Elapsed time:      36 secs
  Files Expected:    5
  Files Restored:    5
  Bytes Restored:    3,221,302,956 (3.221 GB)
  Rate:              89480.6 KB/s
  FD Errors:         0
  FD termination status: OK
  SD termination status: OK
  Termination:      Restore OK

```

2 Best Practices

While it is technically possible to backup multiple VMs in one Bacula hypervisor plugin backup job (VMware, Hyper-V, RHV, Proxmox, etc), this is not necessarily the best way to perform VM backups. It is strongly recommended that one backup Job is created for each VM being backed up for the following reasons:

- By default, if one of your VMs fails to backup in a “multi-VM” backup job, the main Bacula job will terminate “Backup OK – with warnings.” The JobStatus for jobs that terminate “Backup OK” and “Backup OK – with warnings” are not differentiated in the catalog. They are both ‘T’, so this means that you will have to carefully monitor your backup job logs in case some VM backups fail and pay attention to the JobErrors field in the job summaries.
- To address this issue, there is a plugin option called “abort_on_error” in each of the Bacula hypervisor plugins, which causes Bacula to immediately fail the job as soon as an error is detected while backing up a VM. However, if you use this option, and the backup of VM number 11 in a list of 50 VMs fails, then the whole job will be failed, and VMs 12-50 will not be backed up during that job’s run.
- A 1:1 configuration (one VM backed up per job) means that the “abort_on_error” option will make more sense to enable in each job so you will immediately know when a VM fails to backup since the Bacula job will terminate with a “Backup failed” message and ‘f’ in the catalog for the job.
- With a 1:1 VM/Job configuration, re-running a specific VM backup job is simple to do after the cause of the failure is investigated and fixed.
- In the example about the 50 VMs, without a 1:1 configuration, there is no way to re-run a backup of just the one VM that failed to backup.
- Additionally, with a 1:1 VM/Job configuration, job metrics will have more meaning because each VM will be one job, and you will know to expect a specific number of jobs each night with each job representing one VM.
- With a multi-VM per job configuration, each VM will be backed up “serially”, one at a time, disk by disk, VM by VM. A 1:1 configuration will allow several VM backups to be run concurrently which will reduce the overall time to perform the VM backups. Of course, you will need to pay close attention to SD and ESXi storage and networking resources, and adjust the number of concurrent jobs accordingly.
- For some hypervisors (VMware, Proxmox, etc) Bacula provides automation scripts (eg: scan_datacenter.pl for VMware). These scripts are designed so that they will create 1:1 VM/Job configurations. If you plan to make use of these automation scripts, it is a good idea to already be thinking this way, and having your hypervisor plugin backup configurations in a 1:1 configuration from the beginning.

2.1 Failover Cluster

Backup remains seamless in a Failover Cluster setup, regardless of the hosting node for the VM(s) during backup. As long as the user possesses the appropriate credentials on all nodes, the VMs within the cluster will undergo filtering via include/exclude criteria. Remote VMs will be automatically **Migrated** to the local node, backed up, then restored to its original node. Hyper-V does not permit VM migration between nodes during backup operations.

2.2 Recovery

If for any reason (crash, etc.) a backup is interrupted, some actions can be recovered afterwards by calling the `hyperv-recovery.ps1` script with the recovered VM name as parameter:

```
C:\Program Files\Bacula\plugins>powershell -file hyperv-recovery.ps1 my_VM_name
```

so the VM is restored to its pre-backup state: The backup checkpoint is cleaned, the VM state is restored and if the VM has been migrated pre-backup, it's moved back to its original cluster node.