

Scope: Use Commvault with NexaVM through the NexaVM API Proxy in Red Hat Virtualization/ oVirt compatible mode.

This guide is for implementation, operations, and test teams. It covers API Proxy deployment, Commvault onboarding, backup, restore, and troubleshooting.

Table of Contents

1. Architecture Overview
2. Prerequisites
 - 2.1 Release Package
 - 2.2 Deployment Host
 - 2.3 Target Management Node
 - 2.4 Commvault Environment
 - 2.5 Commvault Agent and Component Matrix
3. Deploy and Validate API Proxy
 - 3.1 Remote Deployment or Upgrade
 - 3.2 Default Paths
 - 3.3 TLS and Certificates
 - 3.4 Service and Port Checks
 - 3.5 API and Certificate Checks
 - 3.6 Account Authentication Check
 - 3.7 Minimum Acceptance Checklist
4. Add the Commvault Hypervisor
 - 4.1 Connection Parameters
 - 4.2 Procedure
5. Backup and Restore Operations
 - 5.1 Create a VM Group
 - 5.2 Run VM-Level Backups
 - 5.3 Browse Recovery Points
 - 5.4 File-Level Recovery
 - 5.5 Full VM Restore
 - 5.6 Application-Aware Backup
6. Troubleshooting
7. API Proxy Operations
 - 7.1 Enable API Request Logging
 - 7.2 Collect Logs
 - 7.3 Clean Up Logs
 - 7.4 Uninstall API Proxy

1. Architecture Overview

Commvault does not call native NexaVM APIs directly. In Commvault, add the platform as a Red Hat Virtualization/ oVirt hypervisor. On the NexaVM management node, API Proxy exposes an oVirt/RHV-compatible endpoint:

```
<management-node-ip-or-vip>:16443
```

Logical flow:

```
Commvault CommServe/ VSA
    |
    | HTTPS 16443, oVirt/RHV API
    v
NexaVM API Proxy
    |
    | NexaVM API
    v
NexaVM management node, primary storage, compute nodes
```

- **During backup:** API Proxy creates a NexaVM volume snapshot, creates a temporary data volume from the snapshot, hot-adds that volume to the Commvault VSA proxy VM, and removes temporary resources after the job finishes. The Commvault VSA/MediaAgent reads the disk data and writes it to the backup repository.
- **During restore:** Commvault uses oVirt/RHV restore APIs to create the target VM, target disks, image transfers, and disk attachments. API Proxy translates those requests into NexaVM VM creation, DataVolume creation, NBD/ImageIO writing, root volume swap, data volume attachment, NIC restoration, and optional VM startup.

2. Prerequisites

2.1 Release Package

API Proxy is delivered as a release package. Typical package contents (`tar.gz`):

File	Purpose
<code>api-proxy.jar</code>	API Proxy executable service package
<code>deploy-api-proxy.sh</code>	Remote deployment, upgrade, and uninstall entry script
<code>install-api-proxy.sh</code>	Management-node install and uninstall script, usually called by the remote deployment script
<code>collect-api-proxy-logs.sh</code>	Collects API Proxy logs, systemd journal, management node logs, and dual-management-node information

File	Purpose
cleanup-api-proxy-logs.sh	Cleans API Proxy logs and journal
deploy/api-proxy-common.sh	Shared deployment script logic

Extract the package:

```
tar -xzf api-proxy-<version>.tar.gz -C /tmp
cd /tmp/api-proxy-<version>
```

2.2 Deployment Host

The deployment host is the Linux shell environment where the remote deployment script runs. It can be a jump host, an operations terminal, a temporary VM, or one of the NexaVM management nodes.

For remote deployment, the deployment host must have:

Item	Requirement
Basic commands	bash , ssh , sshpass , scp , curl , tar
SSH access	Can SSH as root to the entry management node
Release package	The full directory api-proxy-<version> has been extracted

2.3 Target Management Node

Item	Requirement
Java	Java 8 runtime, or specify it with JAVA_BIN / JAVA_HOME
Service management	systemd / systemctl
Health checks	curl
Disk tool	qemu-img is available in PATH , or configure adapter.backup.qemu-img-path
Database cleanup	The mysql client is only required when uninstall needs to remove API Proxy-owned tables. If missing, the script skips database cleanup
Port	16443/tcp is reachable from Commvault CommServe, VSA proxy, and Access Node

2.4 Commvault Environment

Item	Requirement
Commvault version	Version 11 or later is recommended. The recommended version is 11.40.51.
Platform version	Nexasphere 2.0.2 or later is recommended. Cloud 1.4.x or later is recommended.
Supported platform storage types	Local storage, NFS, distributed storage (nSDS), and SAN.
Backup limitation	Backup operations are not supported for encrypted virtual machines or disks, such as those using vTPM.
Hypervisor type	Red Hat Virtualization/ oVirt
Virtual Server Agent	The VSA access node/proxy must be deployed inside the managed NexaVM platform and must be schedulable by Commvault
MediaAgent/ Storage Policy	Can write backup data
VM Group	Create protection groups based on business scope
Windows / Linux FLR	Windows and Linux file browse/restore are supported. The protected guest must have the required Commvault agent/components installed. Linux FLR also requires a Linux access node/ FREL
Application-aware	Windows and Linux applications must be configured with the corresponding Commvault plugin, guest agent/components, guest credentials, remote access method, and writer/agent requirements
Time synchronization	CommServe, VSA/Access Node, NexaVM management nodes, and protected guests should have synchronized time

2.5 Commvault Agent and Component Matrix

The following matrix is based on Commvault official oVirt/RHV, VSA, FLR, and application-aware documentation. Ordinary VM-level backup and restore do not require every protected business VM to have a guest agent. Windows/Linux FLR and application-aware backup require the corresponding Commvault agent/components according to restore target, guest OS, and application type.

Quick checklist:

Feature	Required/common Commvault software
VSA access node/proxy	Virtual Server Agent (VSA)

Feature	Required/common Commvault software
Windows FLR	Install Windows File System Agent on the target Windows client/VM, either full or restore-only. The node used to mount/browse the recovery point must have Windows MediaAgent or VSA capability
Linux FLR	Install UNIX File System Agent on the target Linux client/VM, either full or restore-only. Linux file-system browse/mount requires a Linux access node/ FREL; configure Linux MediaAgent+ Virtual Server Agent on the FREL side as required by Commvault
Application-aware	Virtual Server Agent (VSA) on the VSA proxy; Windows File System Agent and the corresponding application iDataAgent/plugin inside the protected Windows guest; UNIX File System Agent and the corresponding database/application agent inside the protected Linux guest; guest OS credentials, application credentials, and the application consistency mechanism

Detailed Matrix:

Scenario	Required components	Deployment location	Notes
Hypervisor onboarding, VM-level backup, full VM restore	Virtual Server Agent (VSA)	Access node/ VSA proxy VM inside the managed NexaVM platform	Deploy at least one access node/ VSA proxy for each hypervisor. This node handles VM backup and restore scheduling, hot-add disk reading, and restore control
RHV/oVirt access node baseline	Red Hat guest tools, Virtual Server Agent	Access node/ VSA proxy VM inside the managed NexaVM platform	The access node/proxy must be in the platform /data center that contains the protected VMs. For Linux proxy nodes, review automatic volume activation settings to avoid HotAdd impact
Windows file-level recovery	Windows File System Agent; Windows MediaAgent or VSA	Windows File System Agent is installed on the target Windows client/VM. Windows MediaAgent/VSA runs on the node used to browse or mount the recovery point	The target Windows client/VM receives restored files. Windows MediaAgent/VSA handles recovery point browse, mount, or restore orchestration. They can be different nodes

Scenario	Required components	Deployment location	Notes
Linux file-level recovery	UNIX File System Agent; File Recovery Enabler for Linux (FREL); Linux MediaAgent+ Virtual Server Agent	UNIX File System Agent is installed on the target Linux client /VM. FREL/Linux access node is used for Linux file-system browse/mount. Linux MediaAgent+ VSA are configured on the FREL/access node side as required by Commvault	The target Linux client/VM receives restored files. FREL/access node identifies, mounts, and browses the Linux file system. They can be different nodes
Application-aware backup	Virtual Server Agent (VSA) on the VSA proxy; File System Agent and corresponding application iDataAgent/plugin inside the protected guest; guest OS credentials; application credentials; application consistency mechanism	Protected Windows /Linux guest; VSA access node/proxy	Windows guests usually require Windows File System Agent and the corresponding application iDataAgent/plugin, relying on VSS /Application Writer. Linux guests usually require UNIX File System Agent and the corresponding database/application agent, with scripts, agent logic, or database consistency mechanisms as required by the application. For Commvault RHV/oVirt, the VSA proxy used for application-aware backup should be a supported Windows VM

Official Commvault documentation:

Topic	Commvault documentation
oVirt/KVM VSA support	Virtual Server Agent Support for oVirt on KVM
RHV access node requirements	System Requirements for Red Hat Virtualization Access Nodes
Install Virtual Server Agent	Installing the Virtual Server Agent

Topic	Commvault documentation
Guest file/folder restore	Restoring Guest Files and Folders
File recovery methods/ Guest Agent	File Recovery Methods
Default Linux FREL	Specifying the Default File Recovery Enabler for Linux
Application-aware requirements	Hypervisor Support and Requirements for Application-Aware Backups
Configure application-aware backup	Configuring application-aware backups for VMs

3. Deploy and Validate API Proxy

API Proxy is installed or upgraded through the remote deployment or upgrade entry script included in the release package. In a dual-management-node environment, the script automatically detects the peer management node and synchronizes the Root CA.

3.1 Remote Deployment or Upgrade

Run the following command from the release package directory on the deployment host:

```
./deploy-api-proxy.sh --install root@<management-node-ip>
```

By default, API Proxy reads the NexaVM configuration from `/usr/local/zstack/apache-tomcat/webapps/zstack/WEB-INF/classes/zstack.properties` on the target management node. If `zstack.properties` is stored elsewhere in the current environment, run the following command on the target management node and identify its absolute path from the output:

```
cloud-ctl status
```

Specify the actual path during deployment with `--zstack-config-path` :

```
./deploy-api-proxy.sh --install \
  --zstack-config-path <absolute-path-to-zstack.properties> \
  root@<management-node-ip>
```

In a dual-management-node environment, the absolute path must exist on both management nodes.

Notes:

| Scenario | What to enter |

| :--- | :--- |

| **Single management node** | Use the only management node IP as `<management-node-ip>` |

| **Dual management nodes** | Use either management node IP. The script detects the other node automatically |

| **Client access address** | Use the management node IP in a single-node deployment. Use the VIP in a dual-node deployment |

Non-interactive example:

```
export API_PROXY_SSH_PASSWORD='<ssh-password>'
export API_PROXY_ASSUME_YES=1
./deploy-api-proxy.sh --install root@<management-node-ip>
```

3.2 Default Paths

Item	Default value
systemd service name	api-proxy.service
Install directory	/opt/api-proxy
Service JAR	/opt/api-proxy/api-proxy.jar
Log directory	/opt/api-proxy/logs
Main log	/opt/api-proxy/logs/api-proxy.log
Upload workspace	/opt/api-proxy/uploads
Runtime workspace	/tmp/api-proxy
TLS directory	/etc/api-proxy/tls
TLS keystore	/etc/api-proxy/tls/server-keystore.p12
HTTPS port	16443
Local health check	https://127.0.0.1:16443/ovirt-engine/api
Root CA download	https://<management-node-ip-or-vip>:16443/ovirt-engine/services/pki-resource

3.3 TLS and Certificates

API Proxy listens on HTTPS by default (port `16443`). If the keystore does not exist, the service generates a self-signed certificate during startup.

When adding the hypervisor in Commvault, you can trust the certificate in the UI or import the Root CA in advance:

```
curl -k https://<management-node-ip-or-vip>:16443/ovirt-engine/services/pki-resource \
-o api-proxy-root-ca.crt
```

For production certificates, prepare a PKCS#12 keystore and make sure the certificate SAN covers the management node IP, VIP, or DNS name used by Commvault to access API Proxy. The default keystore path is:

```
/etc/api-proxy/tls/server-keystore.p12
```

To use another keystore path or password, override the service startup parameters:

```
--server.ssl.key-store=file:/etc/api-proxy/tls/server-keystore.p12
--server.ssl.key-store-password=<password>
```

You can also provide the default keystore password through an environment variable:

```
export ADAPTER_TLS_KEYSTORE_PASSWORD='<password>'
```

In a dual-management-node deployment, both management nodes must use the same /etc/api-proxy/tls and /root-ca.crt / root-ca.key. Otherwise, Commvault may encounter certificate trust issues after VIP failover.

3.4 Service and Port Checks

On the API Proxy management node:

```
systemctl status api-proxy --no-pager -l
systemctl is-active api-proxy
ss -lntp | grep ':16443'
```

Check recent logs:

```
journalctl -u api-proxy -n 100 --no-pager
tail -n 100 /opt/api-proxy/logs/api-proxy.log
```

3.5 API and Certificate Checks

From the API Proxy node:

```
curl -k -i https://127.0.0.1:16443/ovirt-engine/api
curl -k -i https://127.0.0.1:16443/ovirt-engine/services/pki-resource
```

From the Commvault CommServe, VSA proxy, or Access Node network:

```
curl -k -i https://<management-node-ip-or-vip>:16443/ovirt-engine/api
curl -k -i https://<management-node-ip-or-vip>:16443/ovirt-engine/services/pki-resource
```

3.6 Account Authentication Check

Use the NexaVM account that will be configured in Commvault to verify token acquisition:

```
read -r -s MGMT_PASSWORD
curl -k -X POST "https://<management-node-ip-or-vip>:16443/ovirt-engine/sso/oauth/token" \
  -H "Content-Type: application/x-www-form-urlencoded" \
  --data-urlencode "grant_type=password" \
  --data-urlencode "username=<NexaVM-account>" \
  --data-urlencode "password=${MGMT_PASSWORD}"
```

If token acquisition fails, do not treat it as a Commvault problem first. Check the account name, password, tenant/project suffix, and the authentication path between API Proxy and the NexaVM management platform.

3.7 Minimum Acceptance Checklist

Check	Expected result
api-proxy.service	active
16443	Listening and reachable from Commvault
/ovirt-engine/api	Returns the oVirt/RHV-style API root
/ovirt-engine/services/pki-resource	Root CA can be downloaded
Token API	Authentication succeeds with the Commvault credential
Resource discovery	Commvault can enumerate target VMs. Host, Cluster, Storage, Network, Disk, and Tag resources are used for rules, details, backup, and restore selections
First backup	Test VM Full backup completes
First Browse	VM, disk, OVF, or file can be browsed
First restore	Test restored VM can be created, with expected disk and network configuration

4. Add the Commvault Hypervisor

4.1 Connection Parameters

Item	Value
Hypervisor type	Red Hat Virtualization/ oVirt
API address	<management-node-ip-or-vip>:16443
Username	NexaVM/API Proxy authentication account
Password	Password for the above account
Certificate	Trust the API Proxy certificate or import the Root CA
Access Node/ VSA proxy	Select a node deployed inside the managed NexaVM platform. The node must be able to access API Proxy, MediaAgent/backup repository, and the target VM disk hot-add path

4.2 Procedure

1. In Commvault Console, go to the virtualization infrastructure management page.
2. Add a new hypervisor and select **Red Hat Virtualization/ oVirt**.
3. Enter the NexaVM API Proxy address: <management-node-ip-or-vip>:16443
4. Enter the NexaVM/API Proxy authentication account.
5. Select the VSA access node/proxy deployed inside the managed NexaVM platform.
6. Save the hypervisor configuration.
7. Create a VM Group and use **Preview** to confirm that target VMs can be enumerated.
Host, Cluster, Storage Domain, Network/vNIC Profile, Disk, and Tag resources are used in rules, details, backup, and restore selections. The exact display location depends on the Commvault UI page.

If onboarding fails, check the following first:

Check	Method
API Proxy port	From the Commvault side, confirm that <management-node-ip-or-vip>:16443 is reachable. For troubleshooting, run <code>curl -k https://<management-node-ip-or-vip>:16443/ovirt-engine/api</code> to check the API root
Authentication	Use the same account to run the token check or save the hypervisor credential again
Service status	Check API Proxy service and logs on the NexaVM management node
Certificate	Confirm that Commvault trusts the API Proxy Root CA or has accepted the certificate

Check	Method
Port confusion	16443 is the API Proxy endpoint. 443 may be a separate nginx/UI path

5. Backup and Restore Operations

This section describes the main Commvault operations: VM Group creation, VM-level backup, recovery point browse, file-level recovery, full VM restore, and application-aware backup.

5.1 Create a VM Group

A VM Group defines the scope of protected VMs. It is recommended to split VM Groups by business system, operating system, application type, or restore policy.

Common rules:

Rule type	Purpose
VM name	Match VMs by exact or fuzzy name
Guest OS	Separate Linux, Windows, or other systems
Power state	Protect only VMs in a specified power state
Host/ Cluster	Group protection scope by compute host or cluster
Guest Host Name	Match by guest hostname
Tag	Organize protection scope by NexaVM/compatible tags
Include/ Exclude	Include or exclude resources from the rule result

Suggested procedure:

1. Run **Preview** after creating the VM Group.
2. Confirm that the VM count in Preview matches the business expectation.
3. Save the VM Group.
4. Open the VM Group again and confirm that the rules and Preview result can still be read.
5. For the first backup, select a small number of test VMs and run a **Full** backup.

5.2 Run VM-Level Backups

The supported VM-level backup types include Full, Incremental, and Synthetic Full.

5.2.1 Full Backup

1. Select **Backup** on the VM Group or subclient.
2. Select **Full** as the backup type.
3. Submit the job.

4. Monitor parent and child jobs in Job Controller.
5. After the job completes, browse the recovery point and confirm that VM, disk, and OVF metadata are visible.

5.2.2 Incremental Backup

1. Make sure a usable Full recovery point exists for the same VM Group.
2. Select **Incremental** backup.
3. Submit the job and wait for completion.
4. Check the recovery point chain and Browse result.

5.2.3 Synthetic Full

1. Make sure a Full/Incremental backup chain exists.
2. Start **Synthetic Full**.
3. Use job details and recovery point results to confirm completion.

Note: Full, Incremental, and Synthetic Full all enter the same API Proxy snapshot-hot-add control path. Incremental and synthetic behavior is handled by the Commvault backup repository and MediaAgent. Do not judge the final job semantics only by the requested backup type name.

5.3 Browse Recovery Points

After backup completes, browse the recovery point in Commvault.

Object	Expected result
VM	Protected VM is visible
Disk	Disks in the recovery point are visible
OVF/ VM metadata	Configuration metadata for full VM restore is visible
Windows files	Real Windows directories can be enumerated
Linux files	Linux guest file browse is supported

If Browse fails, first check whether the backup job really completed, whether the recovery point is usable, whether the VSA proxy is healthy, and whether Commvault can access the required index and backup repository.

5.4 File-Level Recovery

Windows and Linux file browse/restore are supported. Before FLR, the protected guest must have the corresponding Commvault agent/components installed. Mount behavior, access nodes, and target path permissions differ by OS. Follow the corresponding Commvault FLR workflow for access node and restore target selection.

5.4.1 Windows FLR

Suggested procedure:

1. Browse a completed Windows VM recovery point.
2. Select the target file or directory.
3. Select restore to original location or alternate location.
4. Confirm that the target Windows guest has the required Commvault agent/components installed.
5. Confirm target guest credentials, path permissions, and overwrite policy.
6. Submit the job and verify the restored result.

5.4.2 Linux FLR

Suggested procedure:

1. Confirm that the protected Linux guest has the required Commvault agent/components installed.
2. Confirm that the environment has a Linux access node/ FREL configured.
3. Browse a Linux VM recovery point.
4. Select the target file or directory.
5. Configure restore target and permission handling.
6. Submit the job and verify the restored result.

5.5 Full VM Restore

Full VM restore creates a new VM in NexaVM from a Commvault recovery point. Common scenarios include out-of-place restore, restore to target storage, and network mapping restore.

5.5.1 Pre-Restore Checks

Check	Description
Target cluster/host	Has enough resources to create the VM
Target primary storage	Has enough capacity and can be selected in Commvault
Target network	Network/vNIC Profile can be discovered by Commvault
VM name	Does not conflict with an existing VM
Power-on policy	Restore-time auto-start behavior is clear
Firmware/machine type	BIOS/UEFI and pc/q35 follow restore request and metadata

5.5.2 Procedure

1. Select the recovery point of the target VM in Commvault.

2. Select **Restore Full VM**.
3. Select restore to a new location or specified target location.
4. Set the target Cluster, Storage Domain, and Network/vNIC Profile.
5. Adjust VM name, CPU, memory, and power-on option as needed.
6. Submit the job.
7. Wait for completion in Commvault Job Controller.
8. In NexaVM, confirm the new VM, root disk, data disks, NICs, and running state.

5.5.3 Restore Implementation Path

1. Create placeholder VM
2. Create target disk/DataVolume
3. POST `/imagetransfers` , expose ImageIO/NBD upload target
4. Commvault uploads restore data
5. POST `/diskattachments` , attach root/data disks
6. Root volume swap, NIC restore, optional VM start

5.6 Application-Aware Backup

Application-aware backup is not limited to Windows. Windows and Linux applications both require corresponding Commvault agent/components inside the guest, guest OS credentials, application credentials, remote access method, and an application consistency mechanism. Windows guests usually require Windows File System Agent, the corresponding application iDataAgent/plugin, and VSS /Application Writer. Linux guests usually require UNIX File System Agent, the corresponding database /application agent, and scripts, agent logic, or database consistency mechanisms as required by the application.

5.6.1 Prerequisites

Item	Requirement
Guest agent /components	Windows guests have Windows File System Agent installed. Linux guests have UNIX File System Agent installed. Install the corresponding application iDataAgent/plugin or database/application agent according to the application type
Guest credentials	Can log in to the protected VM. Configure access according to Windows/Linux requirements
Network access	Access Node can reach the guest. Typical Windows checks include WMI, ADMIN\$, firewall, and Remote UAC. Linux checks follow SSH/agent/plugin requirements
Application plugin/agent	Windows applications use the corresponding iDataAgent/plugin. Linux applications use the corresponding database/application agent

Item	Requirement
Application consistency mechanism	Windows checks VSS/Application Writer. Linux follows the application-required agent logic, scripts, freeze/thaw workflow, or database consistency mechanism
Application permission	Backup account has the required application discovery and backup permissions
Name resolution	Access Node, CommServe, and guest can resolve one another consistently

5.6.2 Procedure

1. Enable **Application Aware** on the VM Group or subclient.
2. Configure guest credentials.
3. Confirm that Commvault can discover application instances inside the guest.
4. Start the backup job.
5. Check the VM backup parent job, archive index, application-aware workflow, and application child job.
6. Confirm that the application child job, index, and recovery point status are as expected.

6. Troubleshooting

Symptom	Check first	Suggested action
Commvault cannot add the Hypervisor	16443 , certificate, account, API Proxy service	First access API Proxy from the Commvault side, then check API Proxy logs
Resource discovery is empty	Account permission, NexaVM resource state, API Proxy mapping	Use the same account to confirm VM/Host/Storage /Network visibility
VM Group Preview result is not expected	Rule conditions, case sensitivity, include/exclude order, Tag	Narrow the rule scope and test one condition at a time
Backup fails during snapshot stage	NexaVM volume snapshot, primary storage state, VM/volume state	Check NexaVM tasks and API Proxy snapshot call logs
Backup fails during hot-add stage	VSA proxy state, target host, temporary DataVolume	Check whether the VSA proxy can attach and read the temporary disk
Browse fails	Job completion state, index, backup repository, recovery point state	Confirm that parent and child jobs are complete, then check MediaAgent/index

Symptom	Check first	Suggested action
Full VM restore fails	Target storage capacity, network mapping, VM name, disk format	Locate the restore phase: VM creation, disk creation, image transfer, or disk attachment
Windows FLR fails	Guest credentials, target path permission, recovery point filesystem recognition, index state	Confirm Browse first, then check target Windows guest access and restore path
Linux FLR fails	Linux access node/ FREL, recovery point filesystem recognition, target path permission	Confirm FREL availability, then run Browse and Restore again
Application-aware fails	Guest remote access, application plugin, application permission, writer/agent, child job	Prove Access Node can reach the guest first, then check application discovery, snapshot, and child job according to the Windows/Linux plugin

7. API Proxy Operations

7.1 Enable API Request Logging

By default, full API request and response bodies are not logged. Enable this temporarily when troubleshooting Commvault and API Proxy protocol interaction:

```
./deploy-api-proxy.sh --install --enable-api-logging root@<management-node-ip>
```

After troubleshooting, redeploy without this option to avoid keeping excessive request details.

7.2 Collect Logs

Collect logs for single-management-node or dual-management-node deployments:

```
./collect-api-proxy-logs.sh root@<management-node-ip>
```

Specify an output directory:

```
./collect-api-proxy-logs.sh --output-dir /tmp/api-proxy-logs  
root@<management-node-ip>
```

When aligning with Commvault logs, record:

| Information | Purpose |

| :--- | :--- |

| **Commvault Job ID** | Align API Proxy request timeline |

| **VM name and UUID** | Locate VM, volume, snapshot, and restore session |

| **Failed phase** | Distinguish authentication, resource discovery, snapshot, hot-add, Browse, image transfer, and disk attachment |

| **Job start/failure time** | Extract API Proxy, NexaVM MN, and Commvault logs for the same time window |

7.3 Clean Up Logs

Clean API Proxy logs:

```
./cleanup-api-proxy-logs.sh root@<management-node-ip>
```

Keep only the last 7 days of journal:

```
./cleanup-api-proxy-logs.sh --vacuum-time 7d root@<management-node-ip>
```

7.4 Uninstall API Proxy

Remote uninstall:

```
./deploy-api-proxy.sh --uninstall root@<management-node-ip>
```

Local uninstall:

```
sudo ./install-api-proxy.sh --uninstall
```

By default, uninstall stops and removes `api-proxy.service`, cleans the install directory, log directory, upload directory, runtime workspace, and API Proxy-owned database tables. It does not delete the database or schema itself.

To keep logs or uploads:

```
sudo CLEAN_LOGS=0 CLEAN_UPLOADS=0 ./install-api-proxy.sh --uninstall
```