

# API Proxy Deployment Guide

---

## Table of Contents

---

- [1. Scope](#)
- [2. Release Package Contents](#)
- [3. Environment And Access Requirements](#)
- [4. Default Names And Paths](#)
- [5. Extract The Release Package](#)
- [6. Best Practice: Separate Management And Backup Networks](#)
- [7. Deployment, Installation, Upgrade, And Uninstall](#)
  - [7.1 Remote Deployment](#)
  - [7.2 Local Install](#)
  - [7.3 Post-Deployment Checks](#)
  - [7.4 Uninstall](#)
  - [7.5 Host Data Agent Deployment Options](#)
- [8. Veeam Integration And Worker Deployment](#)
- [9. Entry IP Rules For Single-Management-Node And Two-Management-Node Deployments](#)
- [10. TLS](#)
- [11. Logging And Diagnostics](#)
- [12. Common Environment Variables](#)
- [13. Health Checks And Troubleshooting](#)
- [14. Minimum Acceptance Checklist](#)
- [15. Security Recommendations](#)

## 1. Scope

This guide describes how to deploy, upgrade, uninstall, and troubleshoot the API Proxy service on management nodes. The release package supports deployment on either a single management node or two management nodes. Remote scripts can start from either management node as the entry node and, when required, automatically detect the other management node and relay the operation.

## 2. Release Package Contents

The current release package is distributed as a tar.gz archive:

- `api-proxy-<version>.tar.gz`

After extraction, the directory mainly contains:

api-proxy.jar  
 build-info.properties  
 README.txt  
 DEPLOYMENT.zh-CN.pdf  
 DEPLOYMENT.en-US.pdf      deploy-api-proxy.sh  
                          collect-api-proxy-logs.sh  
 cleanup-api-proxy-logs.sh      install-api-proxy.sh  
 deploy/api-proxy-common.sh  
 data-agent/host-data-agent-linux-amd64  
 data-agent/host-data-agent-linux-arm64  
 data-agent/install-data-agent.sh      data-agent/install-data-agent-to-hosts.sh  
 data-agent/build-info.properties

#### Included files:

| File                                      | Purpose                                                                                                                                   |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| api-proxy.jar                             | Prepared executable service package.                                                                                                      |
| deploy-api-proxy.sh                       | Remote deployment, upgrade, and uninstall entry script.                                                                                   |
| install-api-proxy.sh                      | Local install and uninstall script for a target management node.                                                                          |
| collect-api-proxy-logs.sh                 | Collects service logs, systemd journals, management node logs, and dual-management-node diagnostics.<br>Cleans service logs and journals. |
| cleanup-api-proxy-logs.sh                 | Shared deployment logic, including SSH, relay, and dual-management-node                                                                   |
| deploy/api-proxy-common.sh                | topology detection.<br>x86_64 host data-agent binary.                                                                                     |
| data-agent/host-data-agent-linux-amd64    |                                                                                                                                           |
| data-agent/host-data-agent-linux-arm64    | ARM64 host data-agent binary.                                                                                                             |
| data-agent/install-data-agent.sh          | Local data-agent installer for a compute host, used by automatic and manual installation flows.                                           |
| data-agent/install-data-agent-to-hosts.sh | Manual batch installer for pre-installing or refreshing data-agent on target compute hosts.                                               |
| data-agent/build-info.properties          | Build information for the data-agent payload.                                                                                             |
| build-info.properties                     | Version, build commit, and build time.                                                                                                    |
| DEPLOYMENT.zh-CN.pdf                      | Chinese deployment guide PDF.                                                                                                             |
| DEPLOYMENT.en-US.pdf                      | English deployment guide PDF.                                                                                                             |
|                                           |                                                                                                                                           |

## 3. Environment And Access Requirements

### 3.1 Deployment Host / Execution Environment

The deployment host is optional. It is only needed when you want to run the remote deployment, uninstall, log collection, or log cleanup commands described below. In practice, it is simply a Linux shell environment from which those scripts are executed against the target management nodes.

- It does not have to be a separately prepared VM.
- It can be a temporary Linux VM, a jump host, an operations workstation, or any management node itself.
- If you already copied the full release directory to the target management node and plan to run `install-api-proxy.sh` locally there, you do not need a separate deployment host.

If you use the remote scripts, the deployment host must have:

- Linux shell environment.
- `bash`, `ssh`, `sshpas`, `scp`, and `curl`.
- for extracting the release package.
- Root SSH access to the entry management node.

### 3.2 Target Management Nodes

Each target management node must have:

- Java 8 runtime, or an executable configured through `JAVA_BIN/JAVA_HOME`.
- `systemd` and `systemctl`.
- `curl` for local health checks.
- `qemu-img` installed in advance and reachable through `PATH` during service startup. If it is installed in a non-standard location, configure `adapter.backup.qemu-img-path` before starting the service.
- `mysql` client only when `uninstall` needs to clean owned service tables; if it is missing, the script skips database cleanup and continues the uninstall.
- Port `16443/tcp` reachable from the backup client or other upstream callers.

## 4. Default Names And Paths

| Item              | Default                                   |
|-------------------|-------------------------------------------|
| systemd service   | <code>api-proxy.service</code>            |
| Install directory | <code>/opt/api-proxy</code>               |
| Service jar       | <code>/opt/api-proxy/api-proxy.jar</code> |

|                        |                                                                               |
|------------------------|-------------------------------------------------------------------------------|
| 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 URL | https://127.0.0.1:16443/ovirt-engine/api                                      |
| Certificate endpoint   | https://<AnyManagementNodeIP-or-VIP>:16443/ovirt-engine/services/pki-resource |

## 5. Extract The Release Package

Only the tar.gz package format is provided:

```
tar -xzf api-proxy-1.0.0.tar.gz -C /tmp
cd /tmp/api-proxy-1.0.0
```

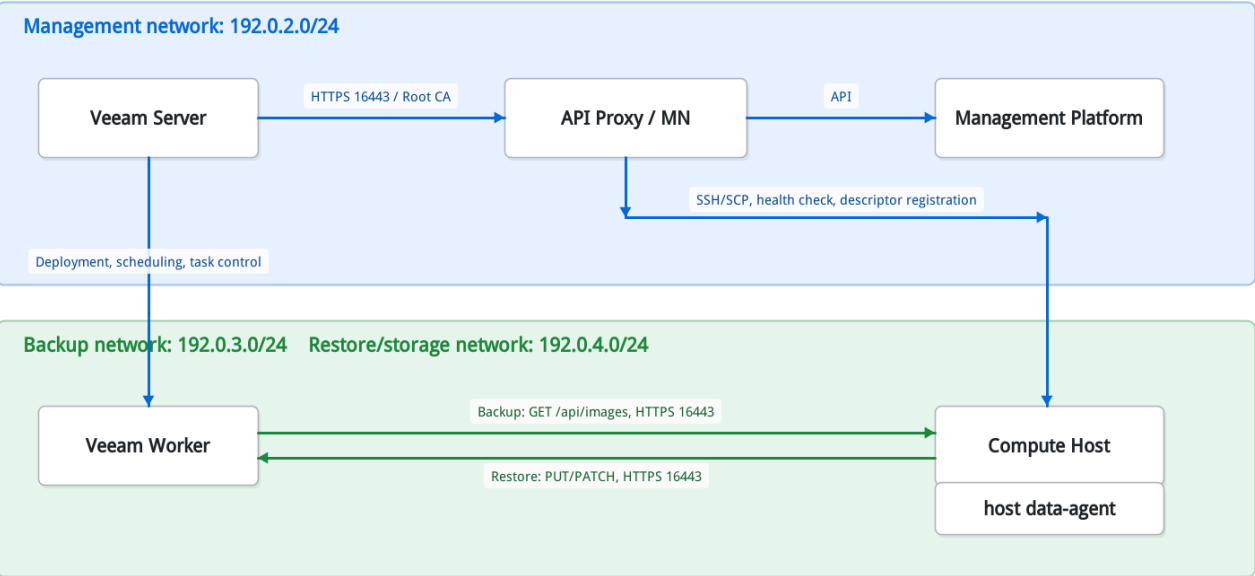
## 6. Best Practice: Separate Management And Backup Networks

For production environments, deploy the management plane and the data plane on separate networks. The management network carries only API, SSH, certificate retrieval, and control-plane requests. The backup network and the restore/storage network carry backup and restore data transfers between Veeam Workers and the target compute-host data-agent. This deployment model prevents backup and restore traffic from being concentrated on the management node and reduces bandwidth pressure on both the MN and the management network.

Recommended topology diagram:

# Recommended Network Topology

Separate the management plane from backup and restore data planes. Veeam Workers access host data-agent directly.



## Recommended logical topology:

| Component               | Management-plane connectivity                                                                                                                  | Data-plane connectivity                                                |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| Veeam Server            | Accesses API Proxy over HTTPS 16443; retrieves the Root CA from /ovirt-engine/services/pki-resource.                                           | Does not carry backup or restore data traffic.                         |
| API Proxy / MN          | Accesses the management-platform API; deploys data-agent over SSH/SCP; accesses data-agent health-check and descriptor-registration endpoints. | Does not proxy backup or restore data traffic.                         |
| Veeam Worker            | Uses API Proxy for Worker deployment, scheduling, and task control.                                                                            | Accesses the target compute-host data-agent directly over HTTPS 16443. |
| Compute Host data-agent | Receives health-check, descriptor-registration, and task-close requests from API Proxy.                                                        | Handles backup read and restore write requests from Veeam Workers.     |

## Recommended traffic paths:

Management plane:  
Veeam Server / Veeam Worker -> API Proxy / MN: HTTPS 16443  
API Proxy / MN -> Management Platform: API  
API Proxy / MN -> Compute Host: SSH/SCP, health check, descriptor registration

Data plane:  
Backup: Veeam Worker -> Compute Host data-agent: HTTPS 16443, GET /api/images  
Restore: Veeam Worker -> Compute Host data-agent: HTTPS 16443, PUT/PATCH

## Recommended network roles:

| Network                  | Recommended traffic                                                                                                                                                                                      | Required connectivity                                                                                                                                           |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Management network       | Veeam adds the platform through API Proxy; API Proxy talks to the management platform; API Proxy deploys data-agent over SSH/SCP; Veeam retrieves the Root CA from /ovirt-engine/services/pki-resource . | Veeam Server/Worker to API Proxy 16443; API Proxy to compute-host SSH port; API Proxy to the management-platform API.                                           |
| Backup network           | Backup download data path. Veeam Workers read disk data from the target compute-host data-agent.                                                                                                         | Veeam Worker to the target host backup-network IP on 16443/tcp; API Proxy to the same data-agent IP on 16443/tcp for health checks and descriptor registration. |
| Restore /storage network | Restore upload data path. Veeam Workers write data to the target compute-host data-agent. This network can be the same as the backup network.                                                            | Veeam Worker to the target host restore /storage-network IP on 16443/tcp; API Proxy to the same data-agent IP on 16443/tcp .                                    |

#### Configuration example:

```
./deploy-api-proxy.sh --install \
--data-agent-enabled \
--backup-network-cidr 192.0.3.0/24 \
--storage-network-cidr 192.0.3.0/24 \
root@<management-node-ip>
```

If backup and restore/storage use different networks, configure them separately:

```
./deploy-api-proxy.sh --install \
--data-agent-enabled \
--backup-network-cidr 192.0.3.0/24 \
--storage-network-cidr 192.0.4.0/24 \
root@<management-node-ip>
```

#### Pre-deployment checklist:

- Confirm that every compute host has both a management-network IP and the expected backup /restore-network IPs.
- Confirm that --backup-network-cidr matches the compute-host IP that should be used for backup downloads.
- Confirm that --storage-network-cidr matches the compute-host IP that should be used for restore uploads.
- From the API Proxy MN, test 16443/tcp reachability to the selected data-agent data-plane IP. API Proxy must reach this address for health checks and descriptor registration.
- From the Veeam Worker network, test 16443/tcp reachability to the selected data-agent data-plane IP. Workers will access this address directly for data transfer.

- In two-management-node deployments, all MNs must share the same `/etc/api-proxy/tls/root-ca.crt` and `root-ca.key`. Manual data-agent pre-install or refresh should be run from the MN whose Root CA is trusted by Veeam, or from another MN synchronized to the same Root CA.
- The data-agent certificate SAN must include the backup/restore IP returned to Veeam. If the network layout or Root CA changes, refresh data-agent on the affected hosts.

Common symptoms:

| Symptom                                            | Common cause                                                                                                                                                                | Recommended action                                                                                                                                                       |
|----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>proxy_url</code> points to the management IP | No CIDR was configured, or the CIDR did not match the target host data-plane IP.                                                                                            | Check <code>adapter.data-agent.backup-network-cidrs / storage-network-cidrs</code> and the target host <code>ip -br addr</code> output.                                  |
| API Proxy fails to create a transfer               | The MN cannot reach the selected data-agent IP, or the data-agent certificate is not trusted by the current MN Root CA.                                                     | From the MN, run <code>curl -k https://&lt;data-agent-ip&gt;:16443/internal/agent/health</code> and check the certificate CA fingerprint.                                |
| Worker reports unknown certificate                 | The Root CA trusted through <code>/pki-resource</code> does not match the CA that issued the data-agent certificate, or the SAN does not include the IP used by the Worker. | Align the MN Root CA, refresh data-agent from the correct MN, and inspect <code>openssl x509 -in data-agent.crt -noout -fingerprint -sha256 -ext subjectAltName</code> . |
| Worker cannot connect to data-agent                | The Worker network cannot reach the target host data-plane IP on <code>16443/tcp</code> .                                                                                   | Check routing, firewalls, security groups, and switching. Temporarily use the management-network CIDR only as a validation fallback.                                     |

## 7. Deployment, Installation, Upgrade, And Uninstall

Two workflows are supported:

- Remote deployment: run `deploy-api-proxy.sh` on the deployment host and let the script install through SSH.
- Local install: copy the full release directory to the target management node and run `install-api-proxy.sh` there.

### 7.1 Remote Deployment

Use the remote deployment entry point included in the release package:

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

Notes:

- In a single-management-node deployment, `<node-ip>` is that management node IP.

- In a two-management-node deployment, <node-ip> can be either management node IP; the script detects the other management node and completes deployment on both nodes.
- If API\_PROXY\_SSH\_PASSWORD is not set, the script prompts for the SSH password without echoing it.
- If an existing deployment is detected, the script prompts for confirmation before overwriting it.
- The installer keeps a backup of the old jar on the target node, refreshes the systemd unit, and restarts the service.
- If a VIP is detected, post-install health checks prefer the VIP endpoint.

Non-interactive example:

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

## 7.2 Local Install

If the full release directory is already on the target management node, run:

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

Notes:

- Local install must run as root.
- Local install does not require a separate deployment host and does not rely on sshpass.

Common local override example:

```
sudo INSTALL_DIR=/data/api-proxy LOG_DIR=/data/api-proxy/logs ./install-api-proxy.sh --install
```

## 7.3 Post-Deployment Checks

Check the API and certificate endpoint:

```
curl -k https://<AnyManagementNodeIP-or-VIP>:16443/ovirt-engine/api
curl -k https://<AnyManagementNodeIP-or-VIP>:16443/ovirt-engine/services/pki-resource
```

Check the service status:

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

## 7.4 Uninstall

Remote uninstall:



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

Local uninstall:

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

Default uninstall behavior:

- Stops and removes `api-proxy.service`.
- Removes the install directory, log directory, upload workspace, and runtime workspace.
- Drops owned service database tables.
- Does not remove the database or schema itself.
- In two-management-node deployments, database-table cleanup runs only once on the last node.
- If the target node does not have the `mysql` client, the script skips database cleanup and continues the uninstall.

Keep selected data during uninstall:

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

## 7.5 Host Data Agent Deployment Options

Host data-agent mode is enabled by default, and the release package includes the data-agent/ payload and installer scripts. During API Proxy installation, configure the backup and restore/storage network CIDRs so image transfer URLs can point to the correct host-side data-agent address. `--data-agent-enabled` can be used to explicitly write the enablement option during installation. The management, backup, and restore/storage networks have different roles:

| Network                  | Purpose                                                                                                                                                                                  | Related configuration                                                                                            |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| Management network       | Veeam connects to API Proxy when adding the platform; API Proxy connects to the management platform; API Proxy uses SSH to auto-install or manually install data-agent on compute hosts. | API Proxy connection address, SSH address, first value in <code>--host</code>                                    |
| Backup network           | Backup download data path. Veeam Workers read VM disk data from target compute host data-agent.                                                                                          | <code>--backup-network-cidr / DATA_AGENT_BACKUP_NETWORK_CIDR / adapter.data-agent.backup-network-cidrs</code>    |
| Restore /storage network | Restore upload data path. Veeam Workers write restore data to target compute host data-agent.                                                                                            | <code>--storage-network-cidr / DATA_AGENT_STORAGE_NETWORK_CIDR / adapter.data-agent.storage-network-cidrs</code> |

If backup and restore use the same network in your environment, configure both CIDRs to the same value. If no CIDR is configured or the target host has no matching address, the data-agent URL falls back to the host management IP.

## Remote deployment example:

```
./deploy-api-proxy.sh --install \  
--data-agent-enabled \  
--backup-network-cidr 10.10.0.0/16 \  
--storage-network-cidr 10.20.0.0/16 \  
root@<node-ip>
```

## Local install example:

```
sudo DATA_AGENT_ENABLED=1 \  
DATA_AGENT_BACKUP_NETWORK_CIDR=10.10.0.0/16 \  
DATA_AGENT_STORAGE_NETWORK_CIDR=10.20.0.0/16 \  
./install-api-proxy.sh --install
```

After installation, the package data-agent/ payload is staged under /opt/api-proxy/data-agent/. API Proxy automatically deploys or refreshes data-agent when the first transfer lands on a remote host. If you want to validate networking, certificates, and SSH credentials before production traffic, you can also pre-install or manually refresh data-agent on selected compute hosts.

Manual pre-install or refresh must be run on a management node where API Proxy is installed. Do not run the batch installer directly on ordinary compute hosts, Veeam Workers, or an external jump host, because it depends on the management-node payload under /opt/api-proxy/data-agent and the Root CA files under /etc/api-proxy/tls/root-ca.crt and /etc/api-proxy/tls/root-ca.key.

```
sudo DATA_AGENT_SSH_PASSWORD='<ssh-password>' \  
/opt/api-proxy/data-agent/install-data-agent-to-hosts.sh \  
--admin-token '<token-if-configured>' \  
--host 192.0.2.155,192.0.3.213 \  
--host 192.0.2.241,192.0.3.255
```

## You can also use a hosts file:

```
# First value is the SSH/management IP; following values are certificate SAN IPs  
192.0.2.155 192.0.3.213  
192.0.2.241,192.0.3.255
```

```
sudo DATA_AGENT_SSH_KEY_PATH=/root/.ssh/id_rsa \  
/opt/api-proxy/data-agent/install-data-agent-to-hosts.sh \  
--hosts-file /tmp/data-agent-hosts.txt
```

The manual installer defaults to /opt/api-proxy/data-agent as the payload directory and uses the current management node's /etc/api-proxy/tls/root-ca.crt plus /etc/api-proxy/tls/root-ca.key to issue the host data-agent certificate. The first address in --host is used for SSH and usually belongs to the management network; additional addresses are added to the certificate SAN list and should cover the backup and restore/storage data-agent IPs that Veeam Workers will access.

If API Proxy is configured with `adapter.data-agent.admin-token` or the `DATA_AGENT_ADMIN_TOKEN` environment variable, pass the same token during manual install or refresh through `--admin-token` or the same environment variable.

After installation or refresh, verify from both the management node and the Worker network:

```
curl -k https://<data-agent-ip>:16443/internal/agent/health
```

If `DATA_AGENT_ADMIN_TOKEN` / `adapter.data-agent.admin-token` is configured, include the same token when checking internal endpoints:

```
curl -k https://<data-agent-ip>:16443/internal/agent/health \
-H "X-Data-Agent-Admin-Token: <token>"
```

The batch installer only installs or refreshes data-agent; it does not perform batch uninstall. To remove data-agent from a compute host, copy `install-data-agent.sh` from the release package or `/opt/api-proxy/data-agent/` to the target compute host, then run on that host:

```
sudo bash install-data-agent.sh --uninstall
```

During remote installation on two management nodes, the deployment script installs the entry node first, lets it generate the Root CA, then syncs `/etc/api-proxy/tls/root-ca.crt` and `/etc/api-proxy/tls/root-ca.key` from the entry node to the other management node. It also removes old API Proxy server certificate artifacts on the target node so that node regenerates its server keystore from the shared Root CA on startup. Manual data-agent installation should be run from a management node using that same Root CA. For local manual installation or existing deployments, still make sure both management nodes use the same Root CA. If each management node generates its own Root CA, `/ovirt-engine/services/pki-resource` may return a different CA after VIP failover, and certificates already issued to data-agent will no longer match the Veeam trust chain. If the Root CA differs between the two nodes, align the Root CA first, regenerate the API Proxy server keystore with the shared Root CA or redeploy, and then install or refresh data-agent.

Routing and lifecycle behavior:

- Backup download `proxy_url` points to the target compute host data-agent address selected from the backup network CIDRs.
- Restore upload `proxy_url` points to the target compute host data-agent address selected from the restore/storage network CIDRs.
- If the target host does not have data-agent, or the existing data-agent is unhealthy or has a certificate that does not satisfy the selected address, API Proxy deploys or refreshes it, verifies health/TLS, and then registers the transfer descriptor.
- Dynamically added hosts do not need pre-installation. The first transfer on a remote host loads the KVM host SSH username, port, and password from the ZStack DB, decrypts the password according to `encrypt.enable.password.encrypt` and `encrypt.encrypt.algorithm`, and then deploys data-agent over SSH/SCP from `/opt/api-proxy/data-agent`. `DATA_AGENT_SSH_USER`, `DATA_AGENT_SSH_PORT`, `DATA_AGENT_SSH_PASSWORD`, and `DATA_AGENT_SSH_KEY_PATH` are fallback/override settings when

the DB credential is unavailable. If password SSH is used, sshpass must be installed on the API Proxy node. If the target host is the management node itself, API Proxy keeps using the original management-node imageio path.

- API Proxy signs data-agent server certificates with /etc/api-proxy/tls/root-ca.crt and /etc/api-proxy/tls/root-ca.key by default. The certificate SAN covers the data-agent IP returned to Veeam. Veeam trusts the same Root CA through /ovirt-engine/services/pki-resource; if the certificate chain or SAN is not acceptable, the next transfer triggers redeployment.
- If no address matches the configured CIDRs, the data-agent URL falls back to the host management IP. Fallback to the management-node data path is allowed only when explicitly enabled. In production, ensure Veeam Workers can reach 16443/tcp on both the backup and restore/storage data-agent addresses.
- During transfer finalize, API Proxy asks data-agent to flush and close the corresponding NBD session. If API Proxy exits unexpectedly, data-agent remains running to avoid interrupting active data streams; later transfers health-check and reuse or refresh it.
- Uninstalling API Proxy removes API Proxy files and the management-node /opt/api-proxy/data-agent payload cache. If a data-agent already installed on a compute host must be removed, copy install-data-agent.sh from the release package to that host and run bash install-data-agent.sh --uninstall.

data-agent is implemented as a single Go binary, so compute hosts do not need Java. The first raw backup / restore path does not require qemu-img or qemu-nbd on compute hosts. If qcow2/cow conversion is moved down to data-agent later, qemu-img will be required on the host side. The installer does not remove system-level qemu-img or qemu-nbd packages by default, to avoid affecting other host services.

## 8. Veeam Integration And Worker Deployment

### 8.1 Worker Role

In oVirt / RHV scenarios, Veeam uses Worker nodes to perform data transfer, backup reads, and restore writes. In this environment, Veeam still deploys and manages Workers in the oVirt model, while API Proxy translates relevant oVirt API calls into requests understood by the backend management platform.

From an operational perspective, users still add a virtualization platform, deploy Workers, create backup jobs, and run restore jobs in the Veeam console. The only difference is that the underlying target is the backend management platform exposed through API Proxy rather than native oVirt.

### 8.2 Veeam Connection Parameters

When adding a virtualization platform in Veeam using the oVirt / Red Hat Virtualization type, use the following parameters.

| Veeam field   | Recommended value                                                                  |
|---------------|------------------------------------------------------------------------------------|
| Platform type | oVirt / Red Hat Virtualization                                                     |
| Address       | Any management node IP or the VIP. For two-management-node deployments, the VIP is |

|             |                                                                                                                                                                                                |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | recommended                                                                                                                                                                                    |
| Port        | API Proxy default is 16443. If your Veeam version only supports 443, change API Proxy to 443 or use port forwarding                                                                            |
| Username    | Management-platform account, for example admin. Do not use admin@internal unless that is the actual account in your environment<br>Plaintext password of the above management-platform account |
| Password    | Use the certificate automatically generated by API Proxy. If Veeam prompts for certificate trust, confirm in the UI or import the Root CA                                                      |
| Certificate |                                                                                                                                                                                                |

You can use the following command to verify whether credentials can obtain an oVirt-style token through API Proxy:

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

## 8.3 Pre-Checks Before Worker Deployment

Before deploying Worker, verify:

- Veeam Backup & Replication Server can reach the API Proxy address and port.

- The Worker network can reach Veeam Server, the backup repository, and any management node IP or the VIP.
- Worker must be assigned a static IP manually; do not rely on DHCP auto-assignment.
- The Worker network can reach the image transfer endpoints used during backup and restore.
- If host data-agent is enabled, Workers must be able to reach the target compute host data-agent addresses on both the backup network and the storage network.
- If host data-agent is enabled, verify before deployment that the configured backup and storage CIDRs match the expected IPs on target compute hosts and that the data-agent port is open.
- For virtual machines migrated from VMware, uninstall VMware Tools in advance; otherwise, the VM may encounter a blue screen issue.
- Target primary storage has enough space for the Worker VM and temporary disks.
- The management-platform account has the permissions required for VM query, restore VM creation, disk create or attach, image transfer, and related operations. During implementation, use an admin account first for quick validation.
- Time synchronization is healthy across the Veeam Server, management nodes, and compute nodes.

## 8.5 Add API Proxy In Veeam And Deploy Worker

The following steps are based on Veeam Backup & Replication 13.0.1.180. UI labels may vary by version; use the equivalent oVirt / Red Hat Virtualization entry points where necessary.

### 8.5.1 Add The Virtualization Platform

1. In Veeam console, open the virtualization infrastructure management page.

Veeam Backup and Replication

Home Server

Add Server Edit Server Remove Server Manage Server Rescan Create Veeam Deployment Kit Tools

**Backup Infrastructure**

- Backup Repositories
- External Repositories
- Scale-out Repositories
- WAN Accelerators
- Service Providers
- SureBackup
- Application Groups
- Virtual Labs
- Managed Servers**
- Microsoft Windows
- oVirt KVM

Home Inventory Backup Infrastructure History

3 servers

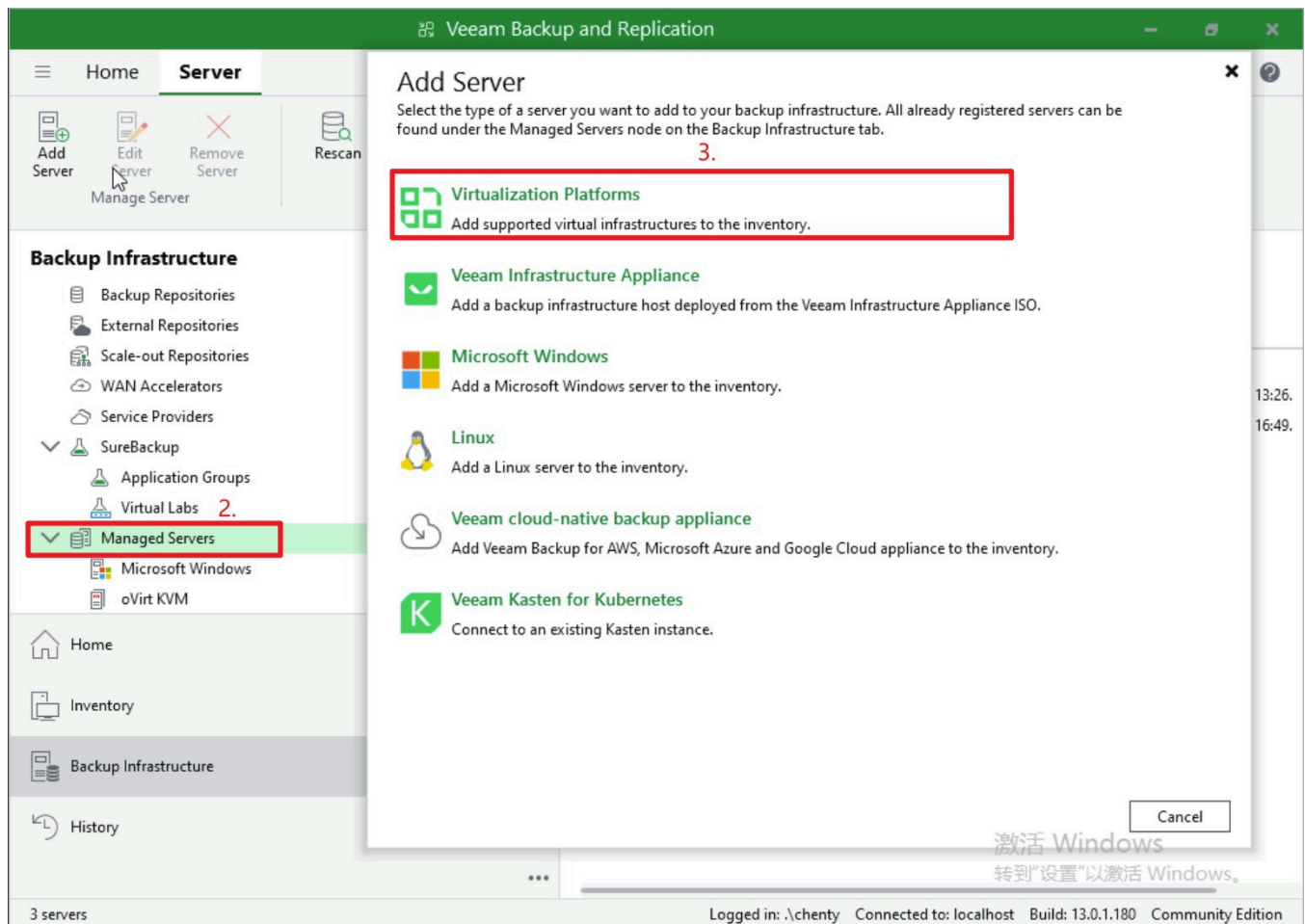
Type in an object name to search for

| Name             | Type                               | Description                              |
|------------------|------------------------------------|------------------------------------------|
| 172-20-19-233    | Microsoft Windows server (defau... | Backup server                            |
| 172.20.19.233    | Microsoft Windows server           | Created by .\chentay at 2026/3/31 13:26. |
| 172.26.50.9:5443 | oVirt KVM Manager                  | Created by .\chentay at 2026/4/25 16:49. |

激活 Windows  
转到“设置”以激活 Windows。

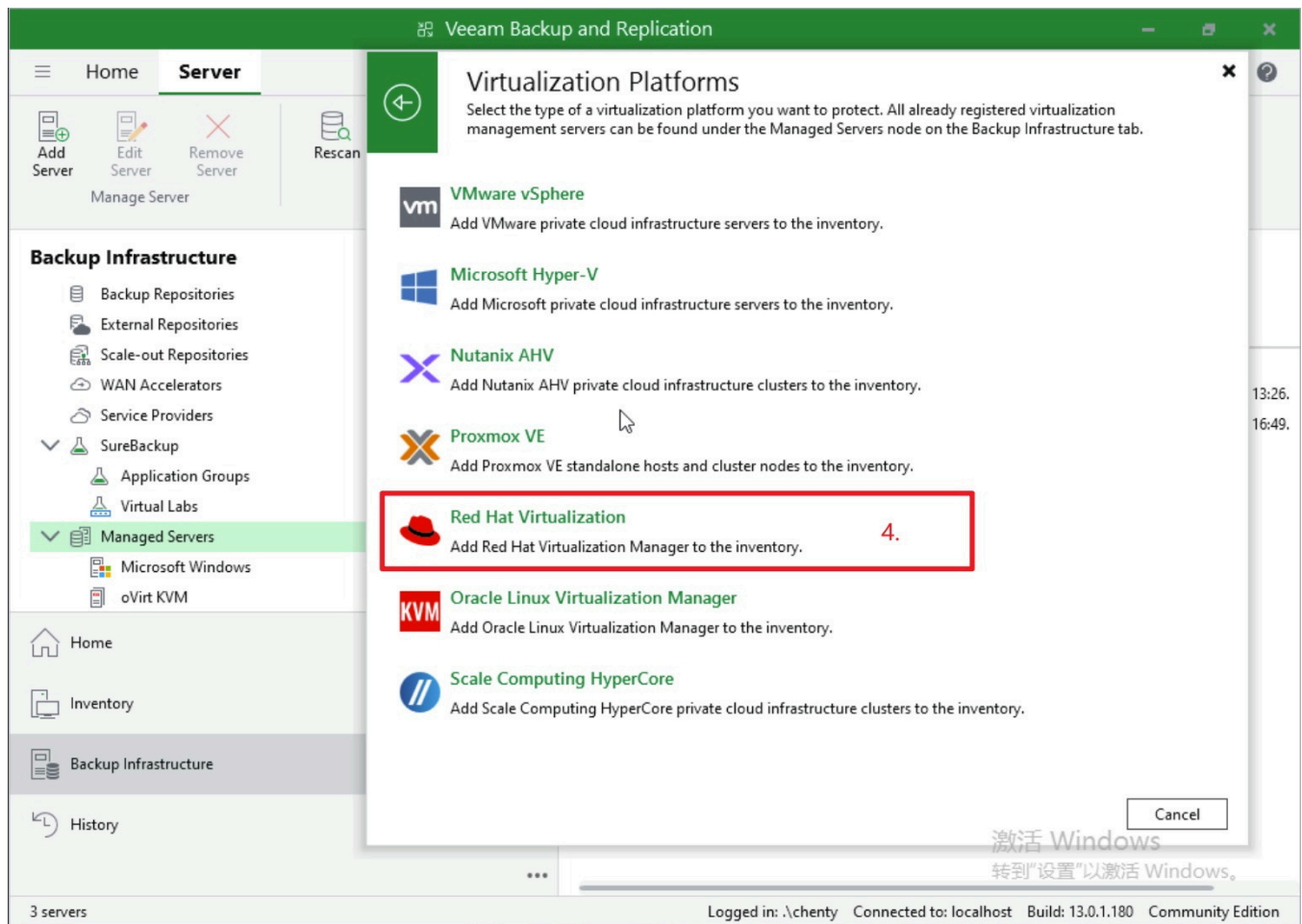
Logged in: .\chentay Connected to: localhost Build: 13.0.1.180 Community Edition

2. Choose to add a virtualization platform.

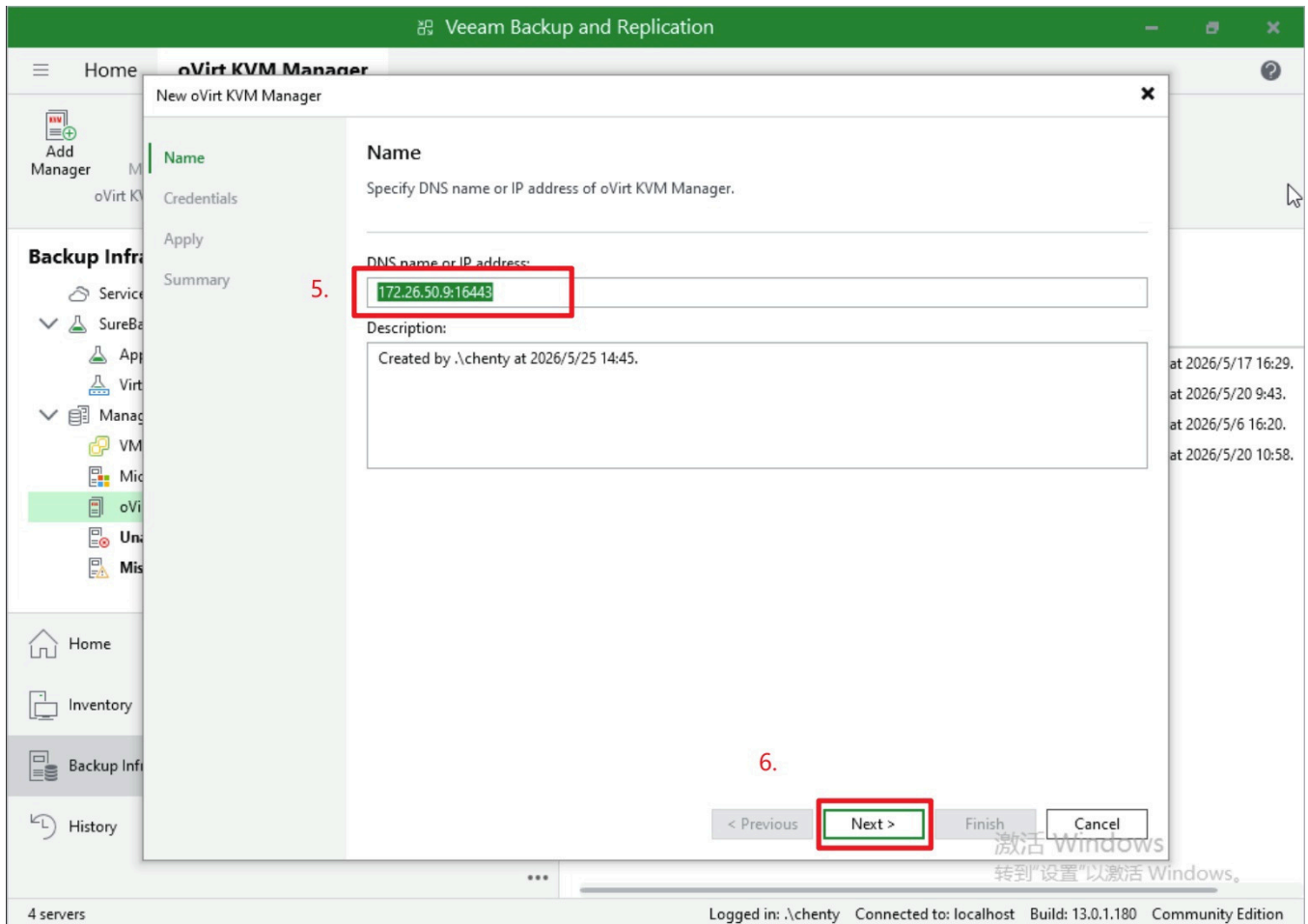


3. Select Red Hat Virtualization / oVirt type.

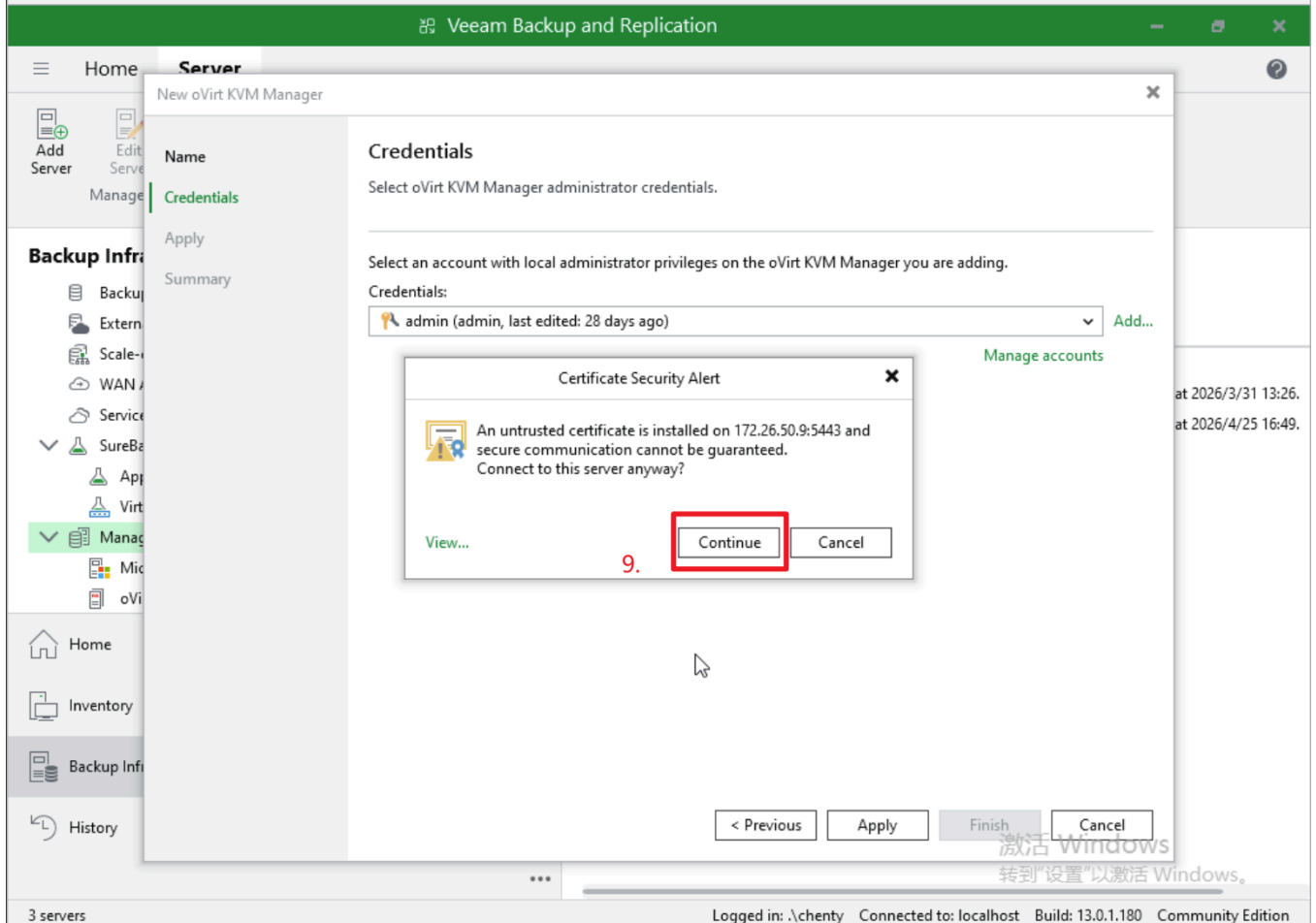
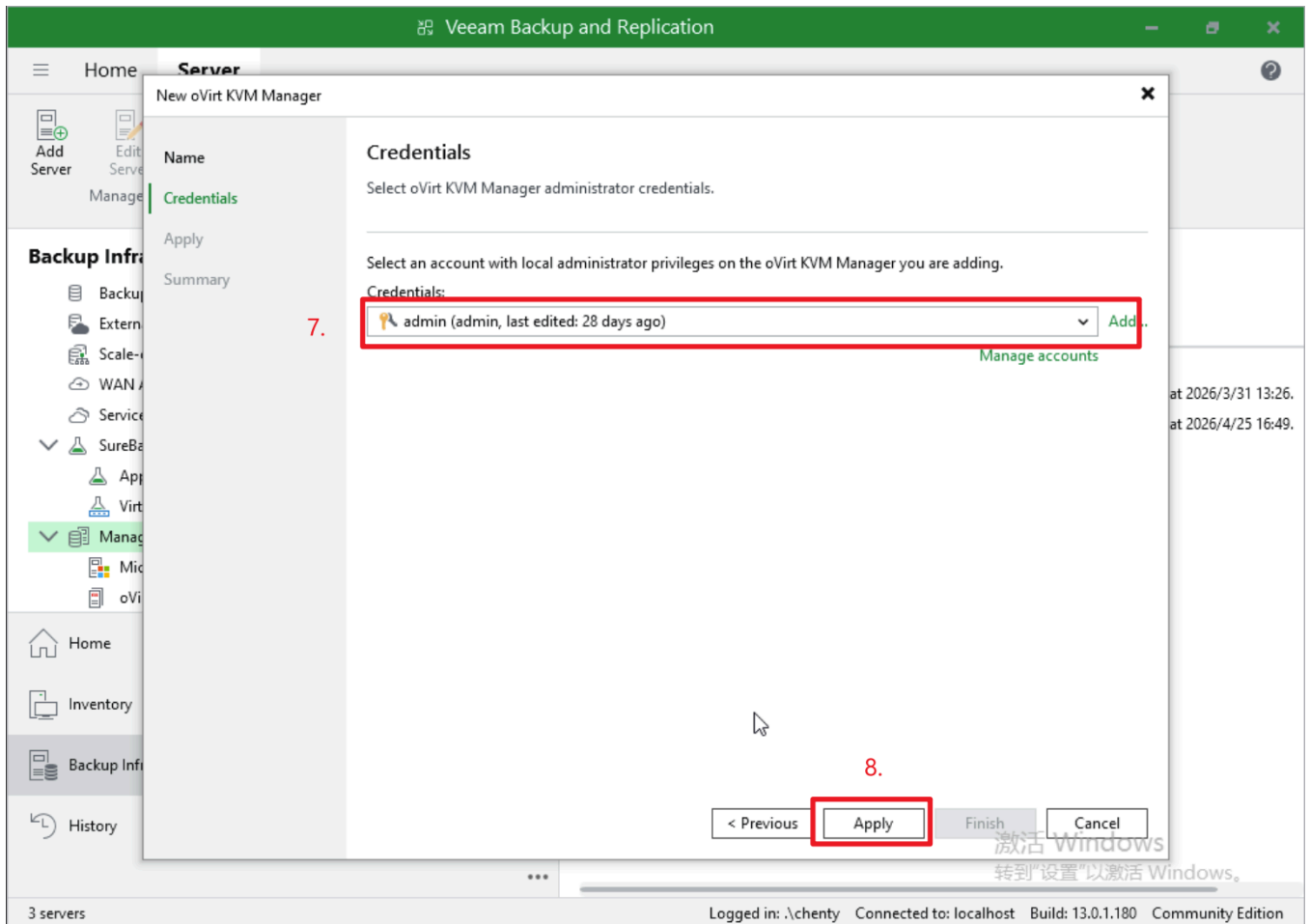




4. Create a new oVirt KVM Manager and fill in the API Proxy address and port.



5. Fill in the management-platform account and password, then trust the API Proxy certificate.



6. Click Finish to complete the oVirt KVM Manager creation.

The screenshot shows the 'New oVirt KVM Manager' wizard in the 'Apply' step. The left sidebar contains a vertical list of steps: 'Name', 'Credentials', 'Apply' (highlighted in green), and 'Summary'. The main area is titled 'Apply' and contains the text 'Please wait while required operations are being performed. This may take a few minutes...'. Below this is a table with two columns: 'Message' and 'Duration'. The table contains two rows of messages, both with green checkmark icons. The first row says 'Successfully registered the oVirt KVM Virtualization Manager'. The second row says 'Successfully refreshed oVirt KVM Virtualization Manager entit...' with a duration of '0:00:01'. At the bottom right, there are four buttons: '< Previous' (disabled), 'Next >' (highlighted with a green border), 'Finish' (highlighted with a red border and the number '10.' above it), and 'Cancel' (disabled). A mouse cursor is visible over the 'Finish' button.

| Message                                                            | Duration |
|--------------------------------------------------------------------|----------|
| ✓ Successfully registered the oVirt KVM Virtualization Manager     |          |
| ✓ Successfully refreshed oVirt KVM Virtualization Manager entit... | 0:00:01  |

### 8.5.2 Create Worker

1. In Veeam console, open the virtualization infrastructure management page.

Veeam Backup and Replication

Home Server

Add Server Edit Server Remove Server Manage Server Rescan Create Veeam Deployment Kit Tools

**Backup Infrastructure**

- Backup Repositories
- External Repositories
- Scale-out Repositories
- WAN Accelerators
- Service Providers
- SureBackup
  - Application Groups
  - Virtual Labs
- Managed Servers**
  - Microsoft Windows
  - oVirt KVM**

Home Inventory Backup Infrastructure History

3 servers

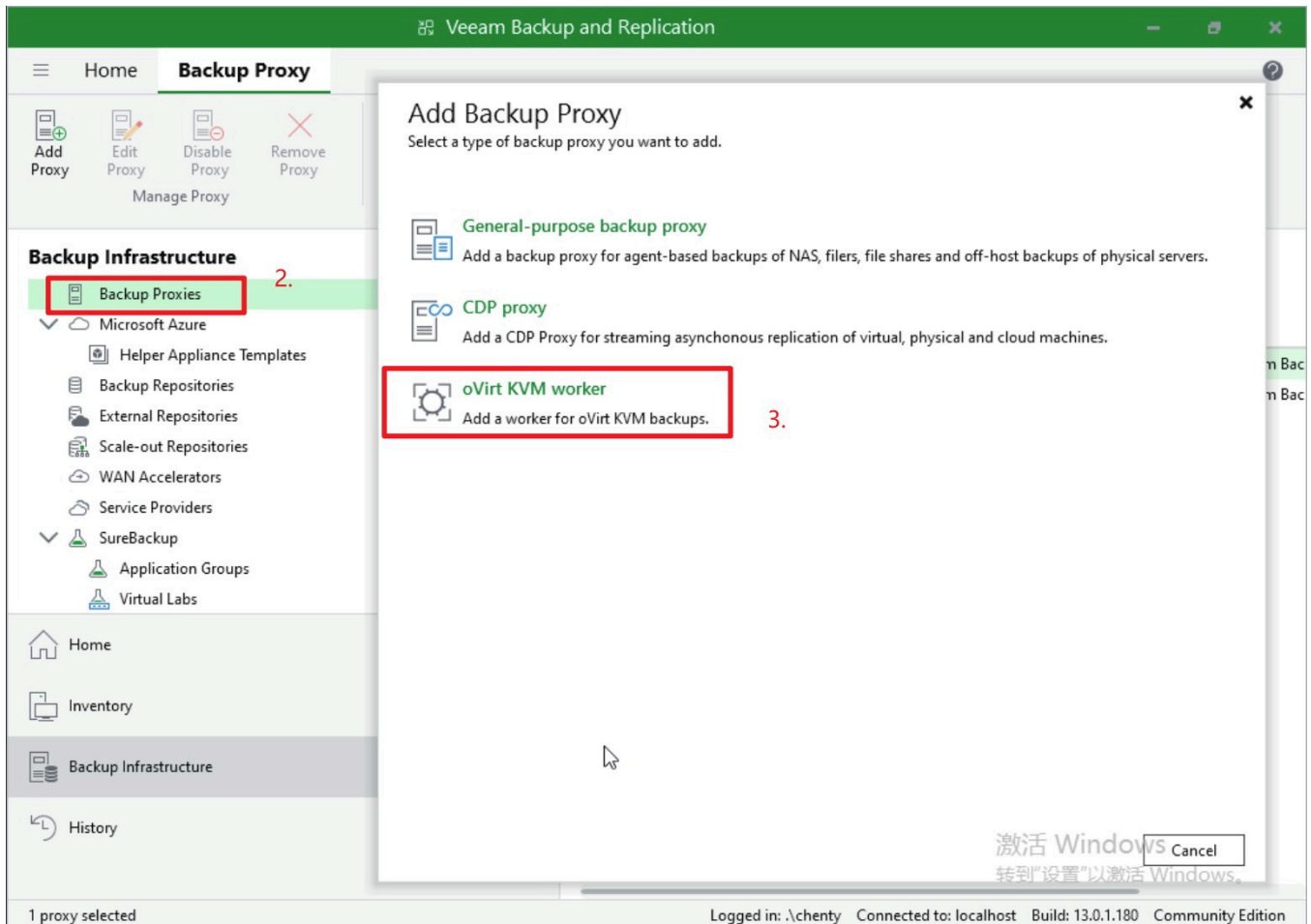
Type in an object name to search for

| Name             | Type                               | Description                             |
|------------------|------------------------------------|-----------------------------------------|
| 172-20-19-233    | Microsoft Windows server (defau... | Backup server                           |
| 172.20.19.233    | Microsoft Windows server           | Created by .\chenty at 2026/3/31 13:26. |
| 172.26.50.9:5443 | oVirt KVM Manager                  | Created by .\chenty at 2026/4/25 16:49. |

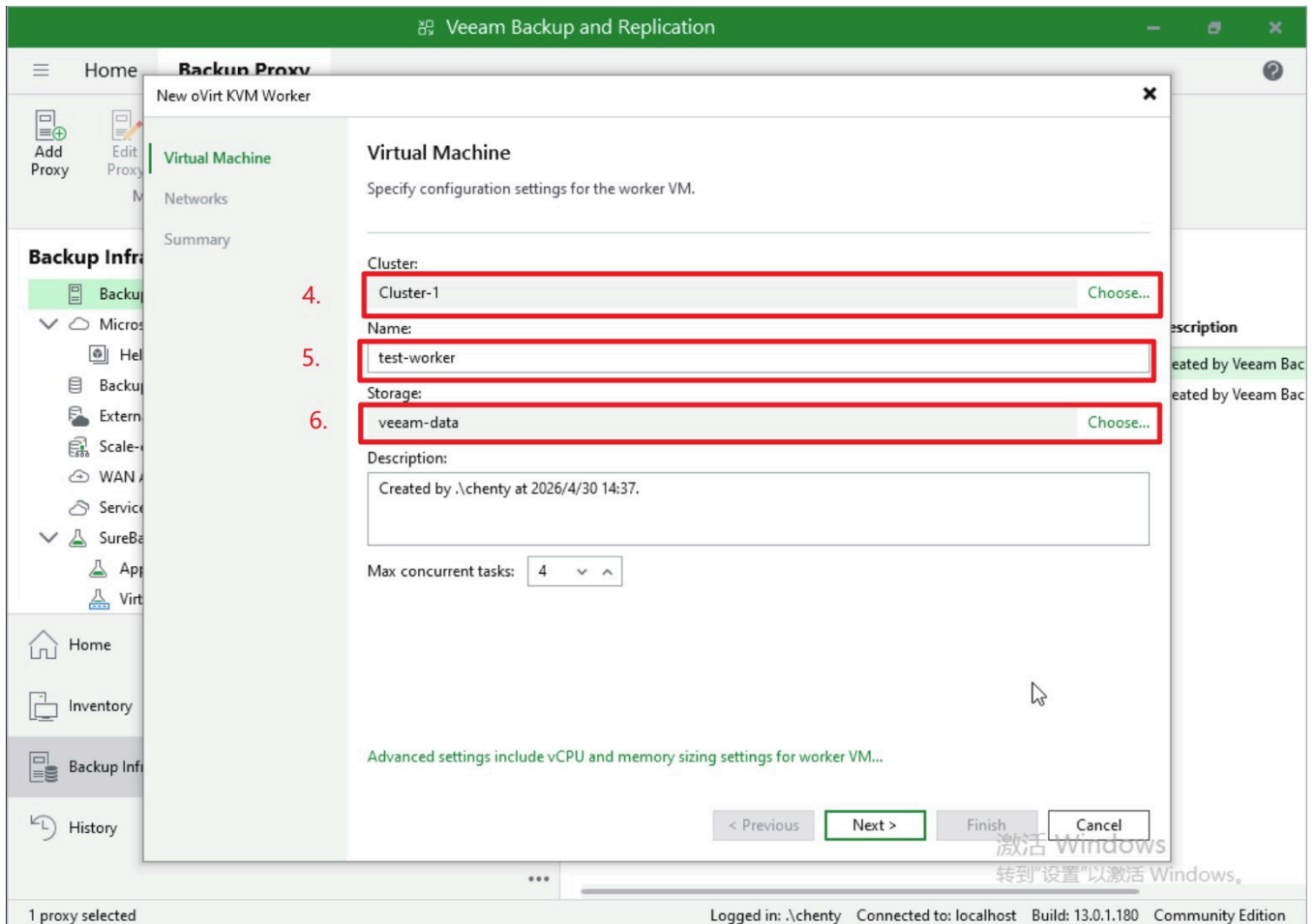
激活 Windows  
转到“设置”以激活 Windows。

Logged in: .\chenty Connected to: localhost Build: 13.0.1.180 Community Edition

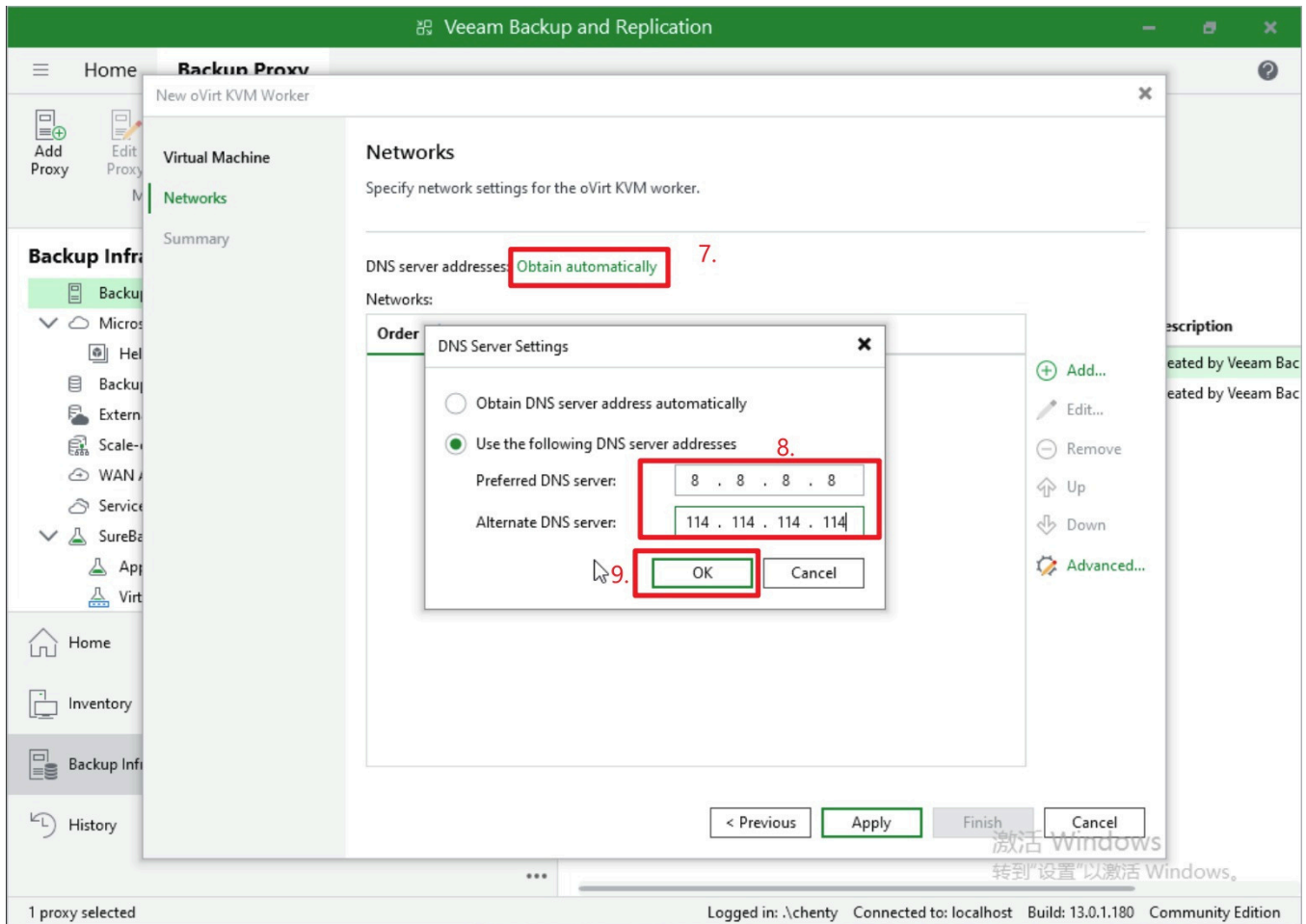
2. Under Backup Proxy, choose to add an oVirt KVM Worker.



3. Configure Worker options including the cluster, Worker name, and storage pool. When selecting the Worker cluster, you must choose Intel hosts, an Intel-only cluster, or a resource scope restricted to Intel hosts by scheduling policy. CPU, memory, and concurrent task settings can start with the Veeam-recommended values and be tuned later based on backup windows.

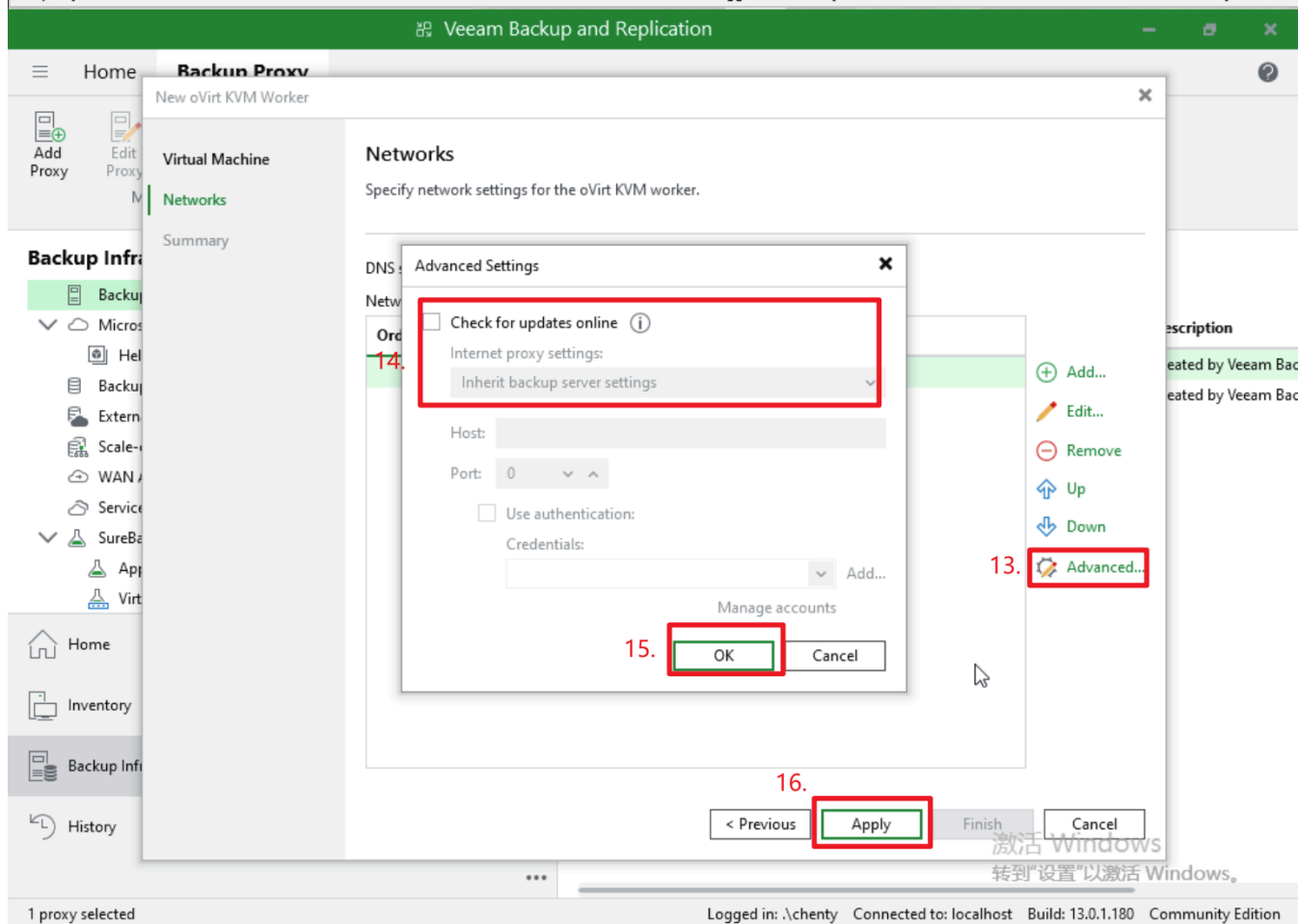
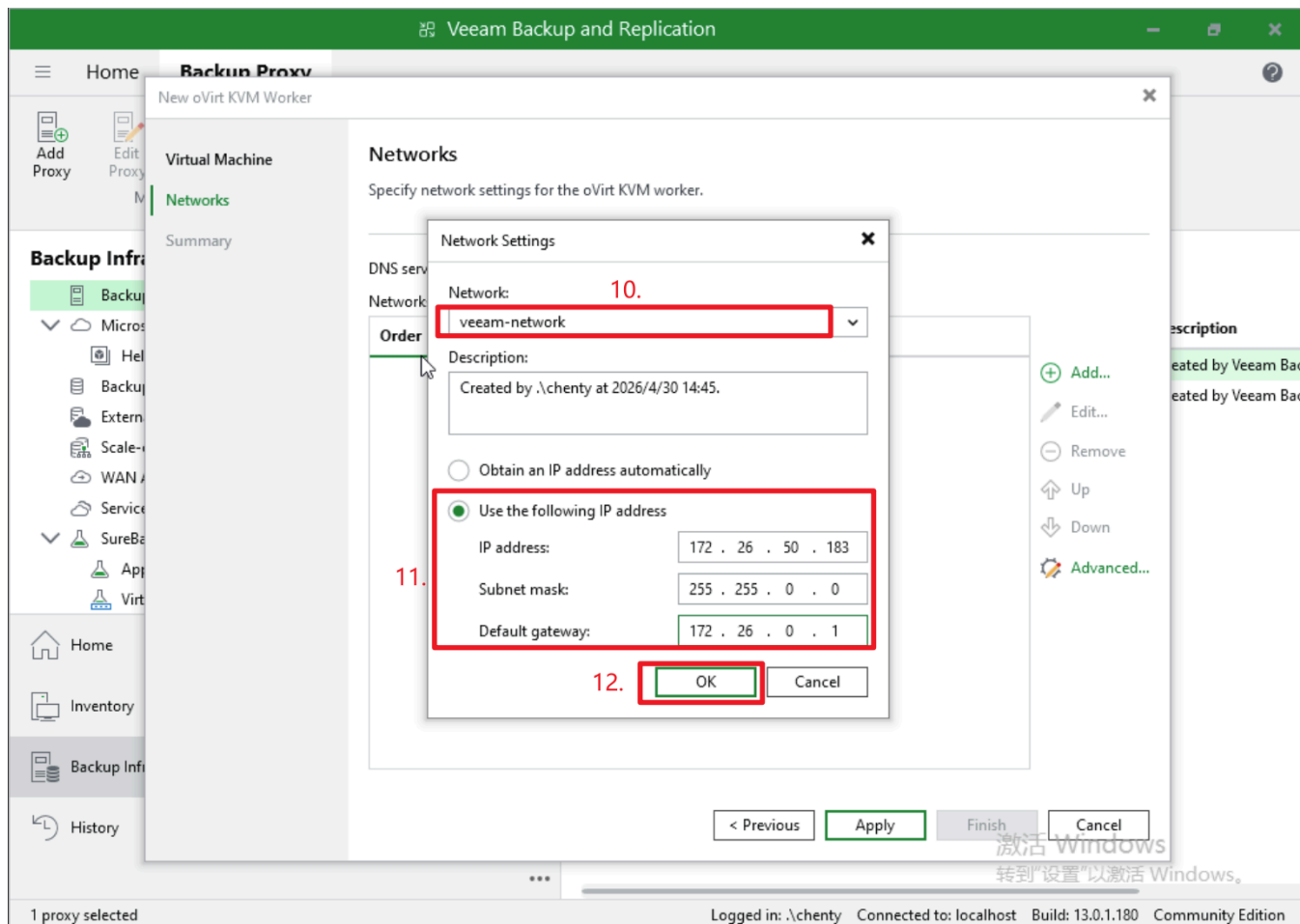


4. Select the Worker management network and ensure it can reach Veeam Server, the backup repository, and the API Proxy address.

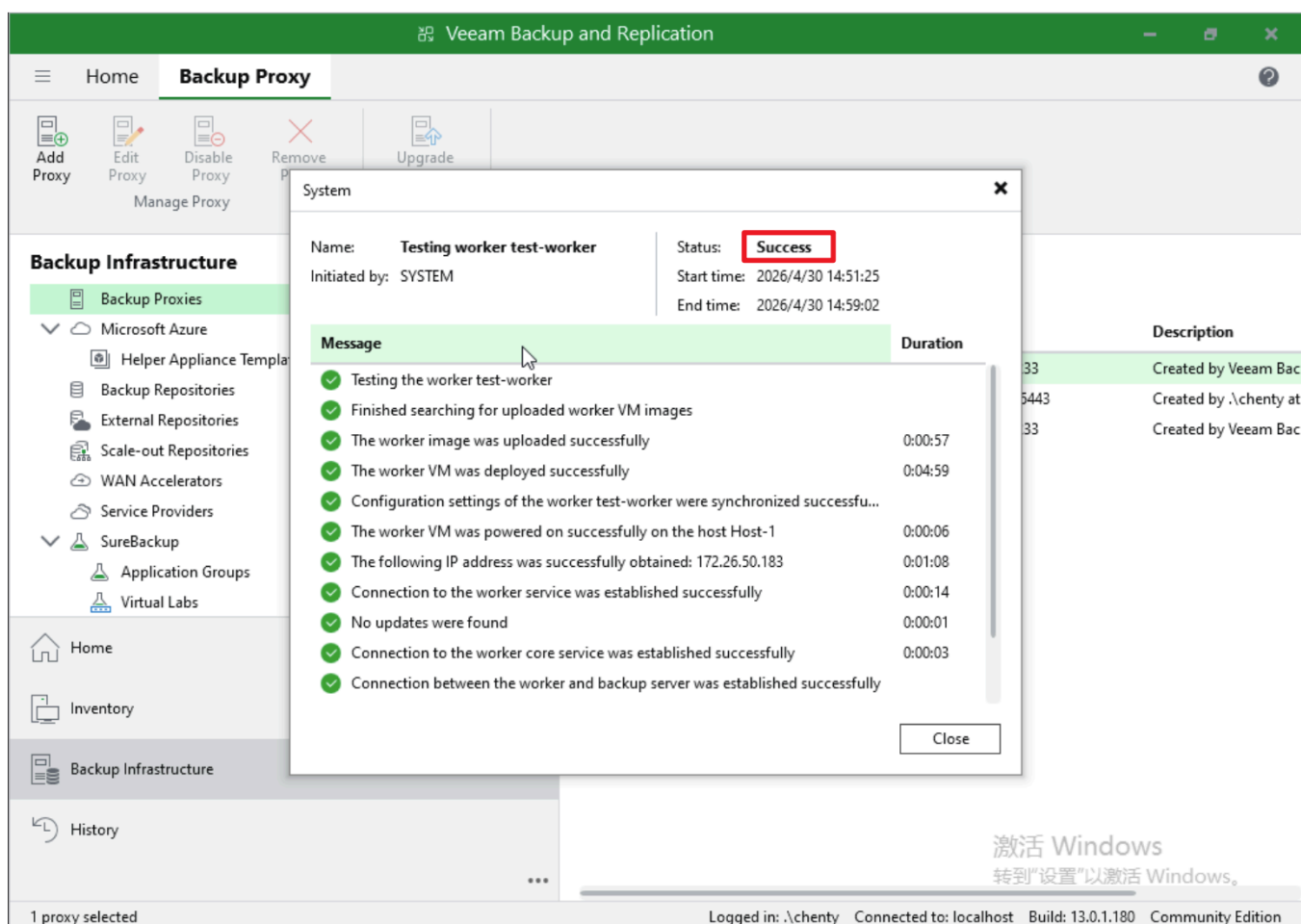
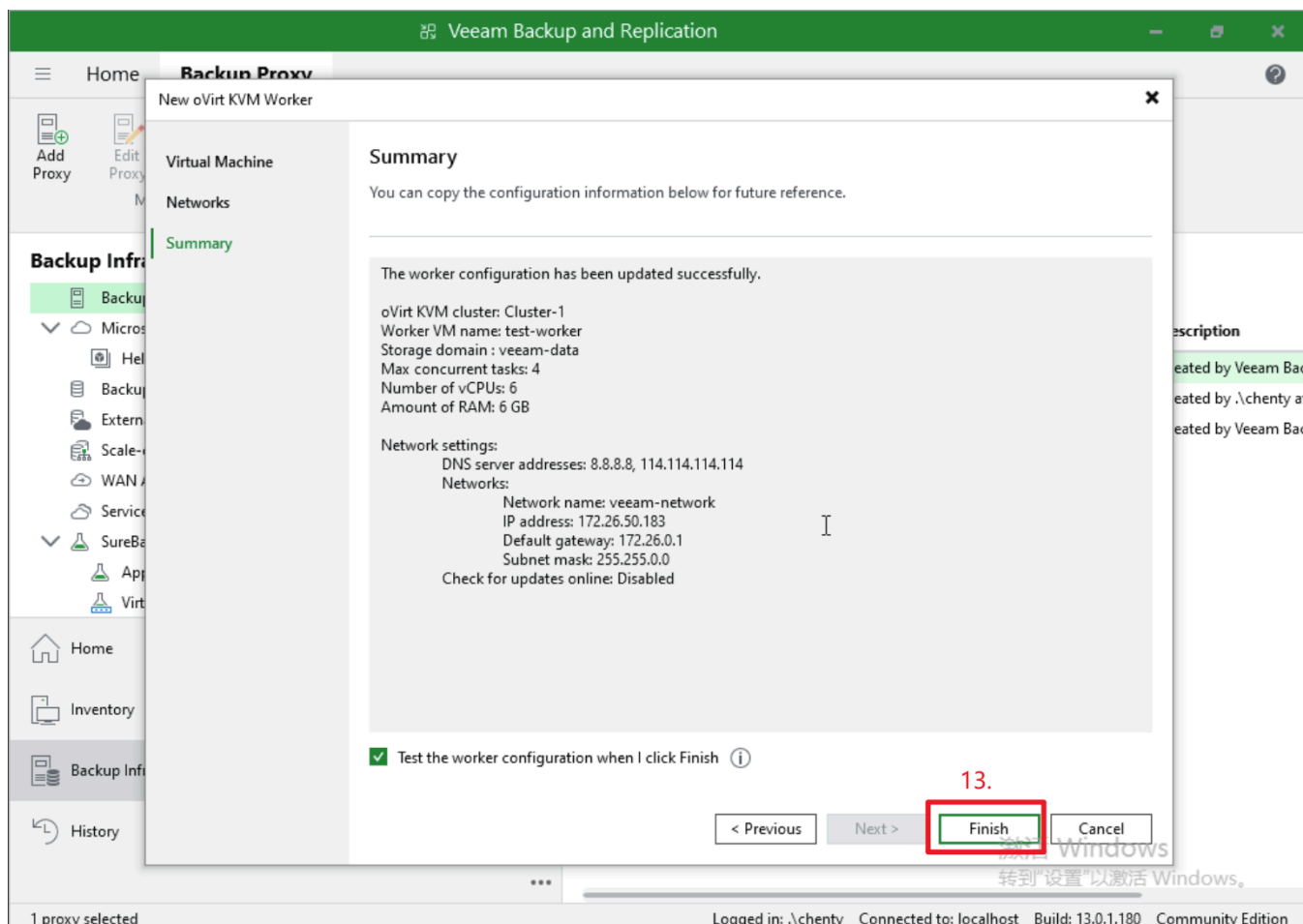


5. Assign a static IP manually for the Worker and ensure it is reachable on the management network and not duplicated by DHCP or other systems.





6. After completing the wizard, confirm in the management-platform console that the Worker VM is created successfully, its IP matches the expected static address, and the VM is running.



7. Back in Veeam console, confirm the Worker status is available.

## 8.6 Troubleshooting Worker Deployment Failures

If Worker deployment or startup fails, prioritize the checks below:

| Symptom                                           | Check item                                                                                                                                                               | Recommended action                                                                                                                                                             |
|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Worker VM cannot start                            | Whether it was scheduled to an AMD host                                                                                                                                  | Delete the failed Worker and redeploy to an Intel host or Intel-only cluster                                                                                                   |
| Worker created but Veeam connectivity is unstable | Whether the correct static IP was manually assigned                                                                                                                      | Reconfigure the fixed static IP and confirm accessibility from Veeam Server, API Proxy, and the repository                                                                     |
| Veeam shows Worker unavailable                    | Worker cannot reach Veeam Server / API Proxy / repository                                                                                                                | Check security groups, firewall, routing, and DNS                                                                                                                              |
| Veeam cannot add the virtualization platform      | API Proxy address, port, certificate, or credentials are incorrect                                                                                                       | Validate with <code>curl -k https://&lt;Address&gt;:16443/ovirt-engine/api</code> and the token endpoint first                                                                 |
| Backup job fails after startup                    | Worker network or storage access issue                                                                                                                                   | Check the Worker network, API Proxy logs, and Veeam job logs                                                                                                                   |
| Transfer creation fails with data-agent           | API Proxy cannot deploy over SSH/SCP, the payload is missing, certificate issuance failed, no matching backup/storage-network IP was found, or agent health check failed | Check <code>ensureDataAgent</code> logs in API Proxy and verify SSH credentials, <code>/opt/api-proxy/data-agent</code> payload, the host data-agent port, and the certificate |
| Data transfer fails with data-agent               | Worker cannot reach host data-agent, or the data-agent certificate is not trusted                                                                                        | From the Worker network, test TCP/TLS reachability to <code>host-data-ip:port</code> and check the certificate SAN; internal health endpoints are for API Proxy only           |
| Restore job fails to                              | Management-platform account permissions are insufficient, or the target cluster, network, or storage selection is incorrect                                              | Re-test with an admin account and confirm the target restore resources are available                                                                                           |

create a VM  
Recommended API Proxy-side log collection:

```
./collect-api-proxy-logs.sh root@<EntryNodeIP>
```

Also export the related Veeam task logs so Veeam request timestamps can be correlated with API Proxy logs.

## 8.7 Backup And Restore Recommendations

- When creating backup jobs in Veeam, select the VMs exposed through API Proxy by following the oVirt / RHV VM selection flow.
- For first-time integration, run an Active Full backup for one test VM to verify the Worker, network, and storage paths.
- During restore, follow the oVirt / RHV restore flow to choose the restore point, target cluster, target storage, and target network.
  - If host data-agent is enabled, first validate one VM backup download and one restore upload to confirm both data-path networks are reachable.
  - In two-management-node deployments, Veeam should use the API Proxy VIP as the connection address to avoid reaching a non-active service entry after a management-node switchover.
- If the API Proxy VIP, management network, or certificate SAN changes, re-check the certificate and Veeam connection settings; regenerate the API Proxy TLS certificate if needed.
  - In production, create a dedicated management-platform account for Veeam and grant only the permissions required for backup and restore. During integration testing, an admin account can be used first to shorten the troubleshooting path.

## 9. Entry IP Rules For Single-Management-Node And Two-Management-Node Deployments

Deployment, uninstall, log collection, and log cleanup all support single-management-node and two-management-node deployments.

<node-ip> denotes the entry node used by the script. It is not always the same as the address clients should use to access API Proxy.

| Scenario               | <node-ip> used in commands                                                                                          | Address clients should use to access API Proxy |
|------------------------|---------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| Single management node | The only management node IP                                                                                         | The same management node IP                    |
| Two management nodes   | Either management node IP; the script uses it as the entry node and automatically detects the other management node | VIP                                            |

- The entry node can be any management node.
- The scripts detect the other management node and the VIP from keepalived configuration.
- During remote operations, the script connects to the entry node first, then relays to the other management node.
- After installation, if a VIP is detected, the script prefers the VIP health endpoint.
- Log cleanup does not stop keepalived and does not intentionally move the VIP.

- If api-proxy is running, log cleanup stops the service briefly, performs cleanup, then starts it again and verifies health.

## 10. TLS

---

The service uses HTTPS by default:

- Listen port: 16443
- Keystore: /etc/api-proxy/tls/server-keystore.p12
- Keystore type: PKCS12

If the keystore is missing, the service startup flow can generate a self-signed certificate through the built-in initialization logic. For production certificates, place the keystore in advance and override the path or password through start arguments.

Example:

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

## 11. Logging And Diagnostics

---

### 11.1 Enable API Request / Response Logging

By default, the service records normal runtime logs but does not record full API request and response bodies. Enable detailed API logging temporarily when troubleshooting protocol interactions:

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

For local install, use the equivalent command:

```
sudo ./install-api-proxy.sh --install --enable-api-logging
```

This writes `--adapter.api-logging.enabled=true` into the systemd start arguments. A later deployment or reinstall without this flag refreshes the unit and returns to the default disabled state.

### 11.2 Log Collection

Collect full logs from a single management node or a two-management-node deployment:

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

Use a specific output directory:

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

## Common options:

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

## Notes:

- In two-management-node deployments, keepalived configuration and journal are collected by default.
- In single-management-node deployments, keepalived data is skipped by default unless you explicitly pass `--keepalived`.
- The default output directory is named `api-proxy-logs-YYYYMMDD-HHMMSS`.
- The top-level output directory also includes `collection-summary.txt` with the entry node, target nodes, VIP, and collection mode.

## Each node directory usually contains:

| File                                | Content                                                                                     |
|-------------------------------------|---------------------------------------------------------------------------------------------|
| <code>summary.txt</code>            | Node status, service state, VIP state, and log directory overview                           |
| <code>api-proxy-logs.tar.gz</code>  | API Proxy file logs                                                                         |
| <code>journal-api-proxy.log</code>  | API Proxy systemd journal                                                                   |
| <code>mn-logs.tar.gz</code>         | Management node file logs                                                                   |
| <code>journal-mn.log</code>         | Management node service journal                                                             |
| <code>keepalived.conf</code>        | Dual-management-node configuration, collected by default in two-management-node deployments |
| <code>journal-keepalived.log</code> | keepalived journal, collected by default in two-management-node deployments                 |

## 11.3 Log Cleanup

By default, the script truncates the current service log, removes rotated compressed logs, and keeps systemd journal for the last 7 days:

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

## Common options:

```
./cleanup-api-proxy-logs.sh --vacuum-time 7d root@<node-ip>  
./cleanup-api-proxy-logs.sh --vacuum-size 1G root@<node-ip>  
./cleanup-api-proxy-logs.sh --skip-journal-vacuum root@<node-ip>  
./cleanup-api-proxy-logs.sh --delete-log-dir root@<node-ip>
```

## Notes:

- If the service is running before cleanup, the script stops api-proxy briefly, performs cleanup, then starts it again and checks health.
- The script does not stop keepalived and does not move the VIP intentionally.
- --delete-log-dir deletes rotated application logs and truncates the active log.
- --skip-failover and --force-active-cleanup are still accepted for compatibility, but they only emit a warning now and do not trigger VIP failover or any keepalived action.

## 12. Common Environment Variables

| Variable                     | Default                                          | Description                                                                       |
|------------------------------|--------------------------------------------------|-----------------------------------------------------------------------------------|
| API_PROXY_SSH_PASSWORD       | empty                                            | SSH password used by remote scripts; prompt if unset                              |
| API_PROXY_ASSUME_YES         | empty                                            | Set to 1 to skip replacement confirmation                                         |
| API_PROXY_ENABLE_API_LOGGING | 0                                                | Set to 1 during remote install to match --enable-api-logging systemd service name |
| SERVICE_NAME                 | api-proxy                                        | Install directory                                                                 |
| INSTALL_DIR                  | /opt/api-proxy                                   | Log directory                                                                     |
| LOG_DIR                      | /opt/api-proxy<br>/logs                          |                                                                                   |
| UPLOAD_DIR                   | /opt/api-proxy<br>/uploads                       | Upload workspace                                                                  |
| RUNTIME_WORK_DIR             | /tmp/api-proxy                                   | Runtime workspace                                                                 |
| JAVA_BIN                     | auto-detect                                      | Java executable path                                                              |
| JAVA_OPTS                    | -Xms256m -Xmx2g                                  | JVM arguments                                                                     |
| APP_OPTS                     | empty                                            | Service start arguments                                                           |
| AUTO_CREATE_UNIT             | 1                                                | Auto-create the systemd unit if it is missing                                     |
| SYSTEMD_UNIT_PATH            | /etc/systemd<br>/system/api-proxy.<br>service    | systemd unit path                                                                 |
| HEALTH_CHECK_URL             | https://127.0.0.1:<br>16443/ovirt-engine<br>/api | Local health check URL                                                            |
| BACKEND_CONFIG_PATH          | auto-detect                                      | Backend configuration path; the install script reads this variable first          |
| MYSQL_BIN                    | mysql                                            | mysql client used during uninstall database cleanup                               |



|                        |             |                                                                                                       |
|------------------------|-------------|-------------------------------------------------------------------------------------------------------|
| API_PROXY_DB_NAME      | auto-detect | Database name used for owned-table cleanup during uninstall; defaults to the current backend database |
| API_PROXY_DB_HOST      | auto-detect | Database host used for owned-table cleanup during uninstall                                           |
| API_PROXY_DB_PORT      | auto-detect | Database port used for owned-table cleanup during uninstall                                           |
| API_PROXY_DB_USERNAME  | auto-detect | Database user used for owned-table cleanup during uninstall                                           |
| API_PROXY_DB_PASSWORD  | auto-detect | Database password used for owned-table cleanup during uninstall                                       |
|                        |             | Remove install directory during uninstall                                                             |
|                        |             | Remove log directory during uninstall                                                                 |
| CLEAN_INSTALL_DIR      | 1           | Remove upload directory during uninstall                                                              |
| CLEAN_LOGS             | 1           | Remove runtime workspace during uninstall                                                             |
| CLEAN_UPLOADS          | 1           | Drop owned service tables during uninstall                                                            |
| CLEAN_RUNTIME_WORK_DIR | 1           |                                                                                                       |
| CLEAN_DATABASE_TABLES  | 1           |                                                                                                       |

## 13. Health Checks And Troubleshooting

Service status:

```
systemctl status api-proxy --no-pager -l
systemctl is-active api-proxy
```

View logs:

```
journalctl -u api-proxy --no-pager -n 200
less /opt/api-proxy/logs/api-proxy.log
```

Check the port and endpoint:

```
ss -lntp | grep ':16443'
curl -k -i https://127.0.0.1:16443/ovirt-engine/api
```

Common issues:

| Symptom                               | Check                                                            |
|---------------------------------------|------------------------------------------------------------------|
| SSH connection fails                  | root login permissions, password, network connectivity, firewall |
| Java missing                          | Install Java 8, or set JAVA_BIN/JAVA_HOME                        |
| Service start fails with qemu-img not | Install qemu-img, ensure it is in PATH, or configure adapter.    |

|                                              |                                                                                                                                                         |
|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| found at startup in logs                     | backup.qemu-img-path                                                                                                                                    |
| Local health check times out                 | Inspect journalctl -u api-proxy and the main log, then verify port, TLS, and database connectivity                                                      |
| Database cleanup is skipped during uninstall | Verify whether the mysql client exists, or provide API_PROXY_DB_* settings explicitly<br>Verify VIP ownership, port reachability, and keepalived health |
| VIP health check fails                       | Verify Intel-host placement, static IP configuration, and network reachability to API Proxy and the repository                                          |
| Veeam Worker remains unavailable             |                                                                                                                                                         |

## 14. Minimum Acceptance Checklist

Before go-live, it is recommended to complete at least the following checks:

- deploy-api-proxy.sh --install completes successfully.
- systemctl status api-proxy shows the service running normally.
- curl -k https://<AnyManagementNodeIP-or-VIP>:16443/ovirt-engine/api returns success.
- Veeam can successfully add API Proxy as oVirt / RHV type.
- Veeam console can discover the clusters, hosts, storage, and VMs exposed through API Proxy.
- Veeam Worker is deployed on an Intel host and shown as available in Veeam.
- At least one test VM backup completes successfully.
- At least one test restore completes, and the restored VM is verified for boot, network, and disk behavior.

## 15. Security Recommendations

- Do not keep API request / response logging enabled longer than necessary; redeploy without --enable-api-logging after troubleshooting.
- Log bundles may contain request paths, resource IDs, stack traces, and environment details. Handle them according to internal procedures before sharing.
- Prefer short-lived environment variables for SSH passwords instead of hard-coding them into scripts or documentation.
- For production certificates, use a keystore issued by a trusted CA and manage the keystore password carefully.