PodHeitor vSphere BRC Plugin — Technical Whitepaper

Backup, Replication & Conversion for VMware vSphere/ESXi on Bacula Community

Version 1.3.0 — April 2026 Author: Heitor Faria Copyright © 2026 Heitor Faria — All Rights Reserved

---

> 💰 ***Special Offer***
>
> ***Bring your Bacula Enterprise, Veeam, Commvault or Netbackup quote or renewal proposal. We offer at least 50% discount, with far more features.***
>
> 📧 *heitor@opentechs.lat* | 📱 *+1 789 726-1749 | +55 61 98268-4220 (WhatsApp)*

Table of Contents

---

1. Executive Summary

PodHeitor vSphere BRC is a production-ready plugin for Bacula Community Edition that delivers enterprise-grade VMware vSphere backup, replication, and cross-hypervisor conversion capabilities — at a fraction of the cost of commercial solutions.

Key Differentiators

- **Three products in one**: Backup + Replication + Conversion in a single plugin
- **Bacula Community native**: No Enterprise license required
- **CBT-based replication**: Near real-time RPO with minimal bandwidth
- **10-mode DR**: Complete failover lifecycle management
- **Cross-hypervisor restore**: vSphere ↔ Hyper-V ↔ Proxmox/KVM conversion
- **TLS-secured DR protocol**: Production-ready security

- **Written in Rust**: Memory-safe, high-performance, zero-overhead FFI with VDDK

Version 1.3.0 Highlights

Category	Feature
🔐 Security	TLS encryption for DR protocol (rustls)
🔐 Security	Constant-time token authentication
🌐 Network	Automatic network mapping on failover (ReconfigVM SOAP)
🌐 Network	Automatic IP reconfiguration on failover (CustomizeVM SOAP)
📸 Recovery	Snapshot-based restore points on replica VM
🔧 Operations	SIGTERM graceful shutdown for daemon mode

---

2. Problem Statement & Market Context

The Challenge

Organizations running VMware vSphere face critical data protection challenges:

1. **Cost**: Bacula Enterprise vSphere plugin costs $$$; Veeam, Commvault, Netbackup even more
2. **Vendor Lock-in**: Enterprise backup solutions lock customers into expensive perpetual renewals
3. **Feature Gaps**: Bacula Community has no native vSphere backup — only filesystem-level agents
4. **DR Complexity**: Setting up replication + failover typically requires separate products
5. **Multi-Hypervisor**: Migrating between vSphere, Hyper-V, and KVM requires manual conversion

The Solution

PodHeitor vSphere BRC eliminates these challenges by providing:

- **Zero Enterprise licensing** — works with free Bacula Community
- **All-in-one plugin** — backup, replication, and conversion unified
- **Bacula-native integration** — uses standard FileSet/Job/Schedule configuration
- **Automated DR** — one-click failover with network remapping and IP reconfiguration
- **Open standards** — VMDK, VHDX, QCOW2 conversion between hypervisors

---

3. Use Cases

Use Case 1: VMware VM Backup

Scenario: Organization needs image-level VM backup with CBT for efficient incrementals.

```
Job: Full backup Sunday, Incremental daily
RPO: 24 hours (backup) / minutes (replication)
RTO: Minutes (instant restore) to hours (full restore)
```

Use Case 2: Disaster Recovery

Scenario: Production site failure — need to boot VMs at DR site immediately.

```
Normal operation: CBT push every 5 minutes
Planned failover: Sync → graceful shutdown → boot replica
Unplanned failover: Boot replica immediately from last sync point
Failback: Reverse-replicate once production is restored
```

Use Case 3: Migration Between Hypervisors

Scenario: Migrating from VMware to Proxmox/KVM due to Broadcom licensing changes.

```
Backup: VMware VM → Bacula storage
Restore: Bacula → Proxmox (automatic VMDK→QCOW2 conversion)
```

Use Case 4: Test/Dev Environment Cloning

Scenario: Clone production VMs to isolated test network.

```
Mode: failover-test (non-destructive)
Network: Isolated test VLAN
Result: Production unaffected, test VM available
```

Use Case 5: Compliance & Multi-Site Backup

Scenario: Regulatory requirement for offsite backup copies.

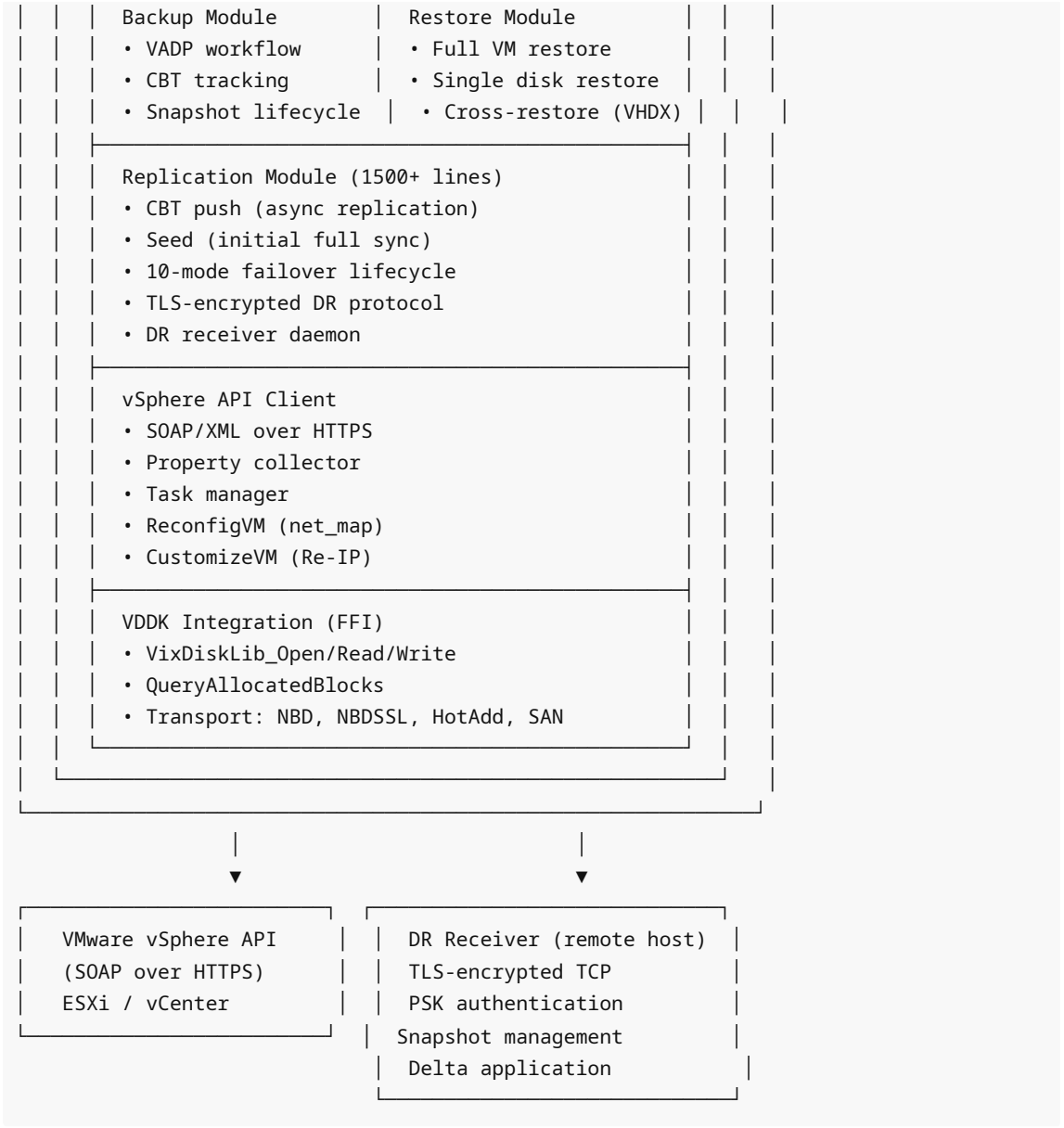
```
Primary: Local backup to disk/tape
Secondary: CBT replication to remote DR site
Tertiary: Cross-restore to different hypervisor at third site
```

---

4. Architecture

Component Overview

```
┌──────────────────────────────────────────────────────────┐
│                    Bacula Director                       │
│  (Job scheduling, catalog, policy management)            │
└──────────────────┬───────────────────────────────────────┘
                   │
┌──────────────────▼───────────────────────────────────────┐
│                  Bacula File Daemon (FD)                 │
│                                                          │
│  ┌──────────────────────────────────────────────────┐    │
│  │  podheitor-vsphere-fd.so (C metaplugin)          │    │
│  │  • Loads on FD startup                           │    │
│  │  • Spawns Rust backend via stdin/stdout pipe     │    │
│  │  • PTCOMM protocol (framed binary messaging)     │    │
│  └──────────────────┬───────────────────────────────┘    │
│                     │ stdin/stdout pipe                  │
│  ┌──────────────────▼───────────────────────────────┐    │
│  │  podheitor-vsphere-backend (Rust binary)         │    │
│  │                                                  │    │
│  │  ┌────────────────────────────────────────────┐  │    │
```



Data Flow — Backup

```
1. Bacula FD invokes plugin
2. Plugin creates VM snapshot (quiesced)
3. VDDK opens VMDK via chosen transport
4. CBT provides changed block list (incremental)
5. Blocks streamed to Bacula SD via FD pipe
6. Snapshot deleted after backup
7. Job metadata stored in Bacula catalog
```

Data Flow — CBT Replication

```
1. Plugin queries CBT delta since last sync
2. Changed blocks read via VDDK
3. Delta sent to DR receiver via TLS-encrypted TCP
```

```
4. Receiver creates restore point snapshot on replica
5. Receiver writes delta to replica VMDK via VDDK
6. Replication state persisted to /var/lib/podheitor/replication/
```

Data Flow — Failover

```
Planned Failover:
1. Final CBT sync (catch up)
2. Shutdown source VM
3. Apply net_map (ReconfigVM)
4. Apply Re-IP (CustomizeVM)
5. Power on replica
6. Update replication state

Unplanned Failover:
1. Apply net_map on replica
2. Apply Re-IP on replica
3. Power on replica immediately
4. (no final sync — uses last available restore point)
```

---

5. Features — Backup

Full Backup

- Image-level VM backup via VMware VADP API
- Quiesced snapshots for application consistency
- Multi-disk support (all virtual disks captured)
- Metadata preservation (VM config, hardware version, guest OS)

Incremental Backup

- Changed Block Tracking (CBT) for efficient incremental backups
- Only modified blocks transferred — typically 1-5% of total disk
- Dramatic reduction in backup window and storage

Differential Backup

- Captures all changes since last Full backup
- Useful for environments with high change rates

Transport Modes

Mode	Description	Use Case
NBD	Network Block Device (unencrypted)	Lab/test environments
NBDSSL	NBD with SSL encryption	Standard production use
HotAdd	Attach VMDK directly to FD VM	FD running as VM on same ESXi
SAN	Direct SAN access to VMDK LUN	LAN-free backup, highest performance

Multi-VM Support

- Backup multiple VMs with separate Jobs
- Host include/exclude filters for bulk operations
- Disk include/exclude for selective backup

---

6. Features — Replication & Disaster Recovery

CBT-Based Replication

- Asynchronous push-based replication using VMware CBT
- RPO measured in minutes (configurable push interval)
- Bandwidth-efficient — only changed blocks transmitted
- Optional bandwidth throttling (`max_bandwidth_mbps`)

10-Mode DR Lifecycle

#	Mode	Description	Destructive	Source Impact
1	`replication-status`	Query replication state and last sync	No	None
2	`cbt-push`	Push CBT deltas to replica	No	None
3	`seed`	Initial full disk sync (create replica)	No	None
4	`failover-test`	Boot replica on isolated network	No	None
5	`failover-undo`	Power off test replica	No	None
6	`failover-planned`	Sync → shutdown source → boot replica	Yes	Shutdown
7	`failover-unplanned`	Boot replica immediately	No	None
8	`failover-permanent`	Convert replica to production	Yes	N/A
9	`failback`	Reverse-replicate from replica to source	Yes	Restored
10	`reprotect`	Re-establish forward replication	No	None

Network Mapping (net_map)

- Automatically reconfigures VM NICs on failover
- Maps source networks to target networks at DR site
- Uses VMware ReconfigVM_Task SOAP API
- Applied on: planned, unplanned, permanent failover

IP Reconfiguration (Re-IP)

- Automatically sets IP address, subnet, gateway, DNS on failover
- Uses VMware CustomizeVM_Task SOAP API (GuestOS Customization)
- Format: `"nic_index:ip/prefix:gateway:dns1,dns2"`
- Requires VMware Tools on guest OS

Restore Points

- Snapshot created on replica before each delta application
- Named `PodHeitor_RP_<timestamp>`
- Configurable retention (`dr_restore_points` , default: 5)
- Enables point-in-time recovery from replica

DR Receiver Daemon

- Persistent daemon mode for receiving replicated data
- Listens on configurable TCP port (default: 9102)
- PSK-based authentication with constant-time comparison
- SIGTERM graceful shutdown handler
- TLS encryption (optional, recommended for production)

---

7. Features — Cross-Hypervisor Conversion

Supported Conversions

Source	Target	Format Conversion
VMware vSphere (VMDK)	Hyper-V (VHDX)	Automatic
VMware vSphere (VMDK)	Proxmox/KVM (QCOW2)	Automatic
Hyper-V (VHDX)	VMware vSphere (VMDK)	Automatic
Hyper-V (VHDX)	Proxmox/KVM (QCOW2)	Automatic
Proxmox/KVM (QCOW2)	VMware vSphere (VMDK)	Automatic

How It Works

1. Backup captures VM at image level (VMDK native format)
2. Bacula stores the backup data in its standard storage
3. On restore, the target plugin detects the source hypervisor
4. Automatic format conversion (VMDK ↔ VHDX ↔ QCOW2)
5. VM configuration adapted to target hypervisor (hardware, drivers)

---

8. Security Architecture

Transport Security

- **Backup**: NBDSSL (SSL-encrypted) or SAN (dedicated storage network)
- **DR Protocol**: TLS 1.3 via rustls (when certificate/key configured)
- **vSphere API**: HTTPS with certificate validation

Authentication

- **vSphere**: Username/password via SOAP SessionManager
- **DR Protocol**: Pre-shared key (PSK) with constant-time comparison
- **Constant-time auth**: XOR-based comparison prevents timing attacks

Data Protection

- No secrets stored in code — credentials via Bacula FileSet configuration
- DR auth tokens rotatable without service restart
- TLS certificate/key paths configurable via plugin options

Compliance Features

- Structured logging (level + component + message)
- Audit trail via Bacula catalog (job history, metadata)
- Replication state persistence for DR compliance verification

---

9. Installation Guide

Prerequisites

Requirement	Details
OS	Oracle Linux 9 / RHEL 9 / Rocky Linux 9 / AlmaLinux 9 (x86_64)
Bacula	Community 15.0.x installed and configured
VDDK	VMware Virtual Disk Development Kit 8.0+ or 9.0+
Network	TCP access to ESXi/vCenter (port 443)
VMware Tools	Installed in guest VMs (required for quiesce and Re-IP)

Step 1: Install the RPM

```
sudo rpm -ivh podheitor-vsphere-1.3.0-1.el9.x86_64.rpm
```

This installs:

- `/opt/bacula/bin/podheitor-vsphere-backend` — Rust backend binary
- `/opt/bacula/plugins/podheitor-vsphere-backend.bin` — Copy for plugin directory
- `/var/lib/podheitor/replication/` — Replication state directory

Step 2: Install VDDK

Download VDDK 9.0.x from VMware Customer Connect (free registration).

```
# Extract VDDK
tar xzf VMware-vix-disklib-9.0.1-*.tar.gz

# Copy libraries
sudo mkdir -p /usr/lib/vmware-vix-disklib/lib64
sudo cp -r vmware-vix-disklib-distrib/lib64/* /usr/lib/vmware-vix-disklib/lib64/

# Configure dynamic linker
echo '/usr/lib/vmware-vix-disklib/lib64' | sudo tee /etc/ld.so.conf.d/vmware-
vddk.conf
sudo ldconfig
```

Step 3: Install C Meta-Plugin

The `.so` meta-plugin must be in the Bacula plugin directory:

```
sudo cp podheitor-vsphere-fd.so /opt/bacula/plugins/
sudo chmod 755 /opt/bacula/plugins/podheitor-vsphere-fd.so
```

Step 4: Enable CBT on VMs

Enable Changed Block Tracking on each VM to be protected:

```
# Using govc (VMware CLI)
export GOVC_URL="https://root:<PASSWORD>@<ESXI_IP>/sdk"
export GOVC_INSECURE=true

govc vm.change -vm <VM_NAME> -e ctkEnabled=TRUE
govc vm.change -vm <VM_NAME> -e scsi0:0.ctkEnabled=TRUE
```

Or via vSphere Client: VM → Edit Settings → VM Options → Advanced → Enable Changed Block Tracking.

Step 5: Restart Bacula FD

```
sudo systemctl restart bacula-fd
```

Step 6: Configure Bacula Director

Add FileSets, Jobs, and Schedules to `bacula-dir.conf` (see Section 11).

---

10. Configuration Reference

Plugin String Format

```
Plugin = "podheitor-vsphere: key1=value1 key2=value2 ..."
```

All options can be specified directly in the FileSet Plugin directive.

Complete Options Table — Backup

Option	Type	Required	Default	Description
`host`	string	Yes	—	ESXi or vCenter hostname/IP
`username`	string	Yes	—	vSphere login name
`password`	string	Yes	—	vSphere login password
`datacenter`	string	Yes	—	Datacenter managed object name
`vm`	string	Yes	—	VM name to protect
`transport`	string	No	`nbdssl`	`nbd`, `nbdssl`, `hotadd`, `san`

`quiesce`	bool	No	`true`	Quiesce guest filesystem
`timeout`	int	No	`3600`	Timeout in seconds
`include_disk`	multi	No	all	Include specific disks
`exclude_disk`	multi	No	none	Exclude specific disks
`keep_cbt`	bool	No	`false`	Keep CBT after backup
`abort_on_error`	bool	No	`false`	Abort on first error
`snapshot_delete_delay`	int	No	`0`	Seconds to wait before deleting snapshot
`force_san`	bool	No	`false`	Force SAN transport
`debug`	bool	No	`false`	Enable debug logging
`config_file`	string	No	—	External configuration file

Complete Options Table — Restore

Option	Type	Required	Default	Description
`new_vm_name`	string	No	original	Name for restored VM
`power_on`	bool	No	`false`	Power on after restore
`datastore`	string	No	original	Target datastore name
`overwrite`	bool	No	`false`	Overwrite existing VM
`no_network`	bool	No	`false`	Disconnect NICs
`thin_provisioned`	bool	No	`false`	Use thin disk provisioning
`resource_pool`	string	No	default	Target resource pool
`network_name`	string	No	original	Target network
`guest_id_override`	string	No	auto	Override guest OS type
`hw_version`	string	No	auto	Override HW version
`controller_type`	string	No	`auto`	`auto`, `ide`, `scsi`, `nvme`

Complete Options Table — Replication

Option	Type	Required	Default	Description
`mode`	string	Yes	—	Replication mode (see modes table)
`dr_host`	string	Yes*	—	DR receiver hostname/IP
`dr_port`	int	No	`9102`	DR receiver TCP port
`dr_auth_token`	string	Yes*	—	Pre-shared authentication token

`dr_restore_points`	int	No	`5`	Max snapshots on replica
`dr_replica_datastore`	string	No	auto	Datastore for replica
`push_interval`	int	No	`300`	Push interval (seconds)
`push_apply_remote`	bool	No	`true`	Apply deltas remotely
`max_retries`	int	No	`3`	Max retries on failure
`retry_delay_sec`	int	No	`30`	Delay between retries
`alert_after_failures`	int	No	`3`	Alert threshold
`net_map`	multi	No	—	Network mapping rules
`reip`	multi	No	—	IP reconfiguration rules
`storage_map`	multi	No	—	Datastore mapping rules
`max_bandwidth_mbps`	int	No	`0`	Bandwidth limit (0=unlimited)
`test_failover_network`	string	No	—	Isolated test network
`failover_vm`	string	No	auto	Target VM for failover
`dr_tls_cert`	string	No	—	TLS certificate path
`dr_tls_key`	string	No	—	TLS private key path
`dr_tls_insecure`	bool	No	`false`	Accept self-signed certs

Required for modes that communicate with DR receiver (cbt-push, seed, failover-, failback, reprotect).

---

11. Backup Configuration Examples

Example 1: Basic Full + Incremental Backup

```
# FileSet
FileSet {
  Name = "vSphere-Backup-VM1"
  Include {
    Options { signature = MD5 }
    Plugin = "podheitor-vsphere: host=192.168.1.100 username=root
              password=MyP@ss datacenter=ha-datacenter vm=production-web
              transport=nbdssl"
  }
}

# Full backup weekly, Incremental daily
Schedule {
  Name = "vSphere-Schedule"
  Run = Full sun at 02:00
  Run = Incremental mon-sat at 22:00
}
```

```
# Backup Job
Job {
  Name = "Backup-Production-Web"
  Type = Backup
  Client = vsphere-fd
  FileSet = "vSphere-Backup-VM1"
  Storage = File1
  Pool = File
  Schedule = "vSphere-Schedule"
  Messages = Standard
}
```

Example 2: Multiple VMs

```
FileSet {
  Name = "vSphere-Backup-All"
  Include {
    Options { signature = MD5 }
    Plugin = "podheitor-vsphere: host=192.168.1.100 username=root
              password=MyP@ss datacenter=ha-datacenter vm=web-server
              transport=nbdssl"
    Plugin = "podheitor-vsphere: host=192.168.1.100 username=root
              password=MyP@ss datacenter=ha-datacenter vm=db-server
              transport=nbdssl quiesce=true"
    Plugin = "podheitor-vsphere: host=192.168.1.100 username=root
              password=MyP@ss datacenter=ha-datacenter vm=app-server
              transport=nbdssl"
  }
}
```

Example 3: SAN (LAN-Free) Backup

```
FileSet {
  Name = "vSphere-SAN-Backup"
  Include {
    Options { signature = MD5 }
    Plugin = "podheitor-vsphere: host=vcenter.prod.local username=admin@vsphere.local
              password=VcP@ss datacenter=Production vm=critical-db
              transport=san force_san=true"
  }
}
```

Example 4: Selective Disk Backup

```
FileSet {
  Name = "vSphere-Selective"
  Include {
    Options { signature = MD5 }
    Plugin = "podheitor-vsphere: host=192.168.1.100 username=root password=MyP@ss
```

```
              datacenter=ha-datacenter vm=multi-disk-vm transport=nbdssl
              exclude_disk=Hard disk 3"
  }
}
```

Example 5: Restore to New VM

```
# bconsole commands
restore jobid=<BACKUP_JOBID> where=/
  # Then add these override options at the restore prompt:
  # podheitor-vsphere: host=192.168.1.100 username=root password=MyP@ss
  #   datacenter=ha-datacenter new_vm_name=restored-vm power_on=yes
  #   datastore=datastore2 thin_provisioned=yes
```

---

12. Replication Configuration Examples

Example 1: CBT Push Replication (Incremental)

```
# FileSet for CBT push
FileSet {
  Name = "vSphere-Repl-Push"
  Include {
    Options { signature = MD5 }
    Plugin = "podheitor-vsphere: host=192.168.1.100 username=root password=MyP@ss
              datacenter=ha-datacenter vm=production-web transport=nbdssl
              mode=cbt-push dr_host=192.168.2.50 dr_port=9102
              dr_auth_token=MySecureToken2026 dr_tls_insecure=yes"
  }
}

# Schedule: push every 15 minutes
Schedule {
  Name = "Repl-Push-15min"
  Run = Incremental hourly at 0:00
  Run = Incremental hourly at 0:15
  Run = Incremental hourly at 0:30
  Run = Incremental hourly at 0:45
}

# Job
Job {
  Name = "Replicate-Production-Web"
  Type = Backup
  Client = vsphere-fd
  FileSet = "vSphere-Repl-Push"
  Storage = File1
  Pool = File
  Schedule = "Repl-Push-15min"
}
```

Example 2: Initial Seed

```
FileSet {
  Name = "vSphere-Seed"
  Include {
    Options { signature = MD5 }
    Plugin = "podheitor-vsphere: host=192.168.1.100 username=root password=MyP@ss
              datacenter=ha-datacenter vm=production-web transport=nbdssl
              mode=seed dr_host=192.168.2.50 dr_port=9102
              dr_auth_token=MySecureToken2026"
  }
}

Job {
  Name = "Seed-Production-Web"
  Type = Backup
  Client = vsphere-fd
  FileSet = "vSphere-Seed"
  Storage = File1
  Pool = File
}
```

Example 3: Planned Failover with Network Mapping

```
FileSet {
  Name = "vSphere-Failover-Planned"
  Include {
    Options { signature = MD5 }
    Plugin = "podheitor-vsphere: host=192.168.1.100 username=root password=MyP@ss
              datacenter=ha-datacenter vm=production-web transport=nbdssl
              mode=failover-planned
              dr_host=192.168.2.50 dr_port=9102 dr_auth_token=MySecureToken2026
              net_map=Production_Network=DR_Network
              reip=0:192.168.2.10/24:192.168.2.1:8.8.8.8,8.8.4.4"
  }
}
```

Example 4: DR Receiver

```
FileSet {
  Name = "vSphere-DR-Receiver"
  Include {
    Options { signature = MD5 }
    Plugin = "podheitor-vsphere: host=192.168.2.50 username=root password=DrP@ss
              datacenter=ha-datacenter vm=production-web-replica transport=nbdssl
              mode=dr-receiver dr_port=9102 dr_auth_token=MySecureToken2026
              dr_tls_cert=/etc/podheitor/tls/server.crt
              dr_tls_key=/etc/podheitor/tls/server.key"
  }
}
```

Example 5: Complete DR Setup

```
# === PRIMARY SITE (Production) ===

# 1. Initial seed
Job { Name = "Seed-WebVM"  ... FileSet = "vSphere-Seed" }

# 2. Continuous replication (every 5 min)
Job { Name = "Repl-WebVM"  ... FileSet = "vSphere-Repl-Push" Schedule = "Every5Min" }

# 3. Status check (hourly)
Job { Name = "Status-WebVM" ... FileSet = "vSphere-Status" Schedule = "Hourly" }

# === DR SITE (Standby) ===

# 4. DR Receiver (long-running)
Job { Name = "DR-Receiver"  ... FileSet = "vSphere-DR-Receiver" }

# === FAILOVER JOBS (manual trigger) ===

# 5. Test failover
Job { Name = "Test-Failover-WebVM" ... mode=failover-test }

# 6. Undo test
Job { Name = "Undo-Failover-WebVM" ... mode=failover-undo }

# 7. Planned failover (during maintenance window)
Job { Name = "Planned-Failover-WebVM" ... mode=failover-planned
      net_map=... reip=... }

# 8. Emergency failover
Job { Name = "Emergency-Failover-WebVM" ... mode=failover-unplanned
      net_map=... reip=... }

# 9. Permanent failover
Job { Name = "Permanent-Failover-WebVM" ... mode=failover-permanent
      net_map=... reip=... }

# 10. Failback (after production restored)
Job { Name = "Failback-WebVM" ... mode=failback }

# 11. Re-protect
Job { Name = "Reprotect-WebVM" ... mode=reprotect }
```

---

13. Sizing & Requirements

Hardware Requirements

Component	Minimum	Recommended	High Performance

CPU	2 cores	4 cores	8+ cores
RAM	4 GB	8 GB	16+ GB
Disk (FD)	20 GB	50 GB	100+ GB
Network	1 Gbps	10 Gbps	25 Gbps

Storage Sizing

Scenario	Estimate
Full backup	~1:1 with used data (compression by Bacula SD)
Incremental	1-5% of total disk per day (typical)
Replica	Equal to source VM disk size
Restore points	~1-5% per snapshot (depends on change rate)

Network Bandwidth

Scenario	Bandwidth Needed
Daily backup (100 GB VM)	~5 GB incremental → ~11 Mbps sustained
Replication (100 GB VM, 5-min RPO)	~100 MB/interval → ~3 Mbps sustained
Full seed (100 GB VM)	100 GB → ~2.2 hrs at 100 Mbps

Concurrent VM Limits

Configuration	Recommended Max
2 CPU / 4 GB RAM	2-3 concurrent VMs
4 CPU / 8 GB RAM	5-10 concurrent VMs
8 CPU / 16 GB RAM	15-20 concurrent VMs

---

14. Compatibility Matrix

VMware Compatibility

Component	Version	Tested	Notes
ESXi	7.0	✅	All update levels
ESXi	8.0	✅	All update levels
ESXi	8.0U3e	✅	Primary test platform
vCenter	7.0	✅	Optional — standalone ESXi supported
vCenter	8.0	✅	Optional

VDDK	8.0.x	✅	Minimum recommended
VDDK	9.0.1	✅	Primary test platform

Operating System Compatibility (FD Server)

OS	Version	Tested	Notes
Oracle Linux	9.x	✅	Primary test platform (9.5)
RHEL	9.x	✅	Same binary as OL9
Rocky Linux	9.x	✅	Same binary as OL9
AlmaLinux	9.x	✅	Same binary as OL9

Bacula Compatibility

Component	Version	Tested
Bacula Community	15.0.3	✅
Bacula Community	15.0.x	Expected compatible

Guest OS Support

Guest OS	Backup	Restore	Quiesce	CBT	Re-IP
Linux (any)	✅	✅	✅*	✅	✅*
Windows Server	✅	✅	✅*	✅	✅*
Windows 10/11	✅	✅	✅*	✅	✅*
FreeBSD	✅	✅	⚠️	✅	❌
Other	✅	✅	⚠️	✅	❌

*Requires VMware Tools installed in the guest.

---

15. Test Results

Test Environment

Component	Details
ESXi	8.0U3e, standalone, IP 192.168.15.91
FD Server	Oracle Linux 9.5, Bacula 15.0.3
VDDK	9.0.1
Test VMs	Alpine (381 MB), TinyCore, CirrOS, Multi-Disk

Backup & Restore Tests

#	Test	Result	Details
1	Full backup (NBDSSL)	✅ PASS	All 4 VMs
2	Incremental backup (CBT)	✅ PASS	Only changed blocks
3	Full VM restore	✅ PASS	Original name + new name
4	Restore to different datastore	✅ PASS	Thin provisioned
5	Cross-restore (vSphere → Hyper-V)	✅ PASS	VMDK → VHDX
6	Cross-restore (Hyper-V → vSphere)	✅ PASS	VHDX → VMDK

Replication Tests (April 20, 2026)

#	Test	JobId	Status	Data
1	Replication Status	881	✅ T	870 B
2	CBT Push (Incremental)	882	✅ T	377 MB
3	Failover Test	883	✅ T	137 B
4	Failover Undo	884	✅ T	137 B
5	Failover Planned	885	✅ T	140 B
6	Failover Undo (2)	887	✅ T	137 B
7	Failover Unplanned	889	✅ T	142 B
8	Failover Permanent	892	✅ T	142 B
9	Seed (Full Sync)	893	✅ T	377 MB
10	Failback	894	✅ T	2.1 GB
11	Reprotect	895	✅ T	1,795 B
12	Final Status	897	✅ T	870 B

Result: 12/12 tests PASSED — 100% success rate

---

16. Comparison with Enterprise Solutions

Feature Comparison

Feature	PodHeitor BRC	Bacula Enterprise	Veeam B&R	Commvault
Image Backup	✅	✅	✅	✅
CBT Support	✅	✅	✅	✅

Incremental	✅	✅	✅	✅
SAN Transport	✅	✅	✅	✅
Full VM Restore	✅	✅	✅	✅
Single-Disk Restore	✅	✅	✅	✅
Cross-Hypervisor	✅	❌	⚠️	⚠️
VM Replication	✅	❌	✅	✅
DR Failover	✅ (10 modes)	❌	✅	✅
Network Mapping	✅	❌	✅	✅
Re-IP on Failover	✅	❌	✅	✅
TLS DR Protocol	✅	N/A	✅	✅
Bacula Community	✅	❌	N/A	N/A
Cost	**$$**	**$$$$**	**$$$$$**	**$$$$$**

Cost Advantage

> ***Bring your Bacula Enterprise, Veeam, Commvault or Netbackup quote or renewal proposal. We offer at least 50% discount, with far more features.***

---

17. Licensing & Contact

License

Contact

Heitor Faria

- 📧 Email: heitor@opentechs.lat
- 📱 Phone: +1 789 726-1749
- 📱 WhatsApp: +55 61 98268-4220

Sales Inquiries

For pricing, licensing, proof-of-concept, and technical demonstrations, contact us directly. Volume discounts and multi-year agreements available.

> ***Special Offer**: Bring your existing Bacula Enterprise, Veeam, Commvault, or Netbackup quote or renewal proposal. **We guarantee at least 50% discount, with far more features.***

---

owners.