



# **Nutanix-Acropolis HyperVisor (Nutanix-AHV) Plugin**

**Bacula Systems Documentation**

---

# Contents

1	Features	2
2	Backup and Restore Strategies	3
3	Installation	6
4	Configuration	7
5	Operations	13
6	Troubleshooting	15
7	Limitations	16

# Contents

---

---

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

---

This document aims at presenting the reader with information about the **Bacula Enterprise Nutanix-AHV Plugin**. The document briefly describes the target technology of the plugin, defines the scope of its operations, and presents its main features.

## 1 Features

Bacula Enterprise is an especially secure and reliable backup and recovery software that is compatible with more databases and hypervisor types than almost any other solution available today. One example is the way it seamlessly integrates with Nutanix AHV to offer an especially powerful backup and recovery solution - even for extremely demanding environments. Nutanix AHV itself is a powerful, Linux KVM based hypervisor, designed in order to make managing computer infrastructure easier. Bacula's Nutanix module is designed to simplify and make efficient the backup and restore procedure of Nutanix VMs.

- Snapshot-based online backup of any guest VM.
- Full, Incremental and Differential block level image backup.
- Backups are consistent at image, disk and host configuration level.
- Ability to restore complete virtual machine image.
- Ability to recover and configure virtual machine network at restore time.
- Ability to restore the data also in raw format over a different filesystem.
- Precise inclusion/exclusion mechanism to control the backup target.
- Automatic backup configuration through hypervisor VMs scan routines.
- Automatic snapshot cleanup processes.
- Compatibility with Deduplication techniques.

- Send backup data to local disk, network disk, block storage, tape or cloud.

**See also:**

Go to:

- [Nutanix-AHV Backup and Restore Strategies](#)
- [Nutanix-AHV Installation](#)
- [Nutanix-AHV Configuration](#)
- [Nutanix-AHV Operations](#)
- [Nutanix-AHV Troubleshooting](#)
- [Nutanix-AHV Limitations](#)

Go back to the [main Nutanix-AHV Plugin page](#).

## 2 Backup and Restore Strategies

This article presents information regarding backup and restore strategies of the Nutanix-AHV Plugin.

### 2.1 Installing Bacula Client on Each Guest

This strategy works by installing a Bacula Enterprise File Daemon on every virtual machine as if they were normal physical clients. In order to optimize the I/O usage on the Nutanix-AHV hypervisor, the user will use Bacula's Schedules, Priorities, and Maximum Concurrent Jobs to spread backup jobs over the backup window. Since all VMs could use the same storage on the Nutanix-AHV hypervisor, running all backup jobs at the same time could create a bottleneck on the disk/network subsystem since Bacula will walk through all filesystems to open/read/close/stat files.

Installing a Bacula Enterprise File Daemon on each virtual machine permits to manage virtual servers like physical servers and also to use all Bacula Enterprise's features such as:

- Quick restores of individual files
- Checksum of individual files for Virus and Spyware detection
- Verify Jobs
- File/Directory exclusion (such as swap or temporary files)
- File level compression
- Accurate backups.

### 2.2 Image Backup With Nutanix-AHV Plugin

With the image backup level strategy, the Bacula Enterprise Nutanix-AHV Plugin will save the Client disks at the raw level, in the Nutanix-AHV context.

Bacula's Nutanix-AHV Plugin will query the guest VM through the hypervisor API to read and save the content of virtual machines disks using snapshots and the native Nutanix Filesystem (NDFS). During backups, Nutanix-AHV Plugin will save the integrity of disks images and also guest VM configurations to allow guest VM restores with their original parameters.

All those operations are handled by an additional proxy VM which is described in the next section.



```

| /@nutanix-ahv/690d74f9-9ce4-45fb-9b23-149afebe18f7_VmName.name
↪ |
| /@nutanix-ahv/690d74f9-9ce4-45fb-9b23-149afebe18f7/conf.1646137248688.conf
↪ |
| /@nutanix-ahv/690d74f9-9ce4-45fb-9b23-149afebe18f7/eef9d485-dba3-4e3f-b828-
↪ 46a8c3412bec.1646137267248.bmp |
| /@nutanix-ahv/690d74f9-9ce4-45fb-9b23-149afebe18f7/eef9d485-dba3-4e3f-b828-
↪ 46a8c3412bec.1646137267248.abmp |
| /@nutanix-ahv/690d74f9-9ce4-45fb-9b23-149afebe18f7/eef9d485-dba3-4e3f-b828-
↪ 46a8c3412bec.1646137267262.bvmdk |
+-----+
↪ -----+

```

## 2.5 Network Restore

The Nutanix-AHV plugin restores network interfaces. The user can influence the process using the two restore plugin parameters `disconnect_network` and `network_address`.

When attaching a new network interface to a guest VM the API alters the creation based on three factors:

- Is the NIC connected to the network?
- Does the NIC requests an ip address?
- If the NIC requests an ip address does it requires a specific one?

By default the Nutanix-AHV plugin does the following:

- It uses the `disconnect_network` argument to know whether to disconnect the NIC or not. The default value is *false*
- If the original guest VM had an ip address, the restored guest VM will request one
- If specified with the `network_address` plugin parameter, it asks for the relevant static address

If this default execution fails, the Nutanix-AHV plugin will try again with a disconnected NIC.

---

**Note:** The Nutanix-AHV plugin allows the request of static IP addresses only on a managed network.

---

### See also:

Go back to:

- [Nutanix-AHV Features](#)
- [Nutanix-AHV Backup and Restore Strategies](#)

Go to:

- [Nutanix-AHV Configuration](#)
- [Nutanix-AHV Operations](#)
- [Nutanix-AHV Troubleshooting](#)
- [Nutanix-AHV Limitations](#)

Go back to the [main Nutanix-AHV Plugin page](#).

## 3 Installation

To use the Nutanix-AHV Plugin, a Bacula File Daemon has to be installed on a VM running in the Nutanix-AHV environment. This VM is referred to as a proxy VM throughout this document. The proxy VM needs to have network access to the Nutanix API. The default network adapter should suffice.

Since all backup interactions are network based, any Bacula Enterprise File Daemon with access to the necessary Nutanix endpoints can be used to run the plugin.

### 3.1 Nutanix-AHV Installation with BIM

In order to install the Nutanix-AHV Plugin with BIM, install the File Daemon with BIM and choose to install the Nutanix-AHV Plugin during the FD installation.

[Click here](#) for more details on the plugin installation process with BIM.

#### See also:

See an alternative way of installing the Nutanix-AHV Plugin - *[Nutanix-AHV Installation with Package Manager](#)*.

Go back to the *[main Nutanix-AHV Plugin Installation page](#)*.

Go back to the *[main Nutanix-AHV Plugin page](#)*.

### 3.2 Nutanix-AHV Installation with Package Manager

#### Prerequisites

The `Plugin Directory` directive of the `File Daemon` resource in `/opt/bacula/etc/bacula-fd.conf` should point to the location where the `nutanix-ahv-fd.so` plugin is installed. The default directory is: `/opt/bacula/plugins`

```
FileDaemon {
  Name = bacula-fd
  Plugin Directory = /opt/bacula/plugins
  ...
}
```

#### Installation Steps

An example for a Debian based Linux package manager would be a configuration file `/etc/apt/sources.list.d/bacula.list` with the following content.

```
# Bacula Enterprise
deb https://www.baculasystems.com/dl/@cust@/debs/bin/@version@/bionic-64/ bionic main
deb https://www.baculasystems.com/dl/@cust@/debs/nutanix/@version@/bionic-64/ bionic
↳nutanix
```

Use `apt-get update` to update the package cache. After that the Plugin can be installed using `apt-get install bacula-enterprise-nutanix-ahv-plugin`

#### See also:

See an alternative way of installing the Nutanix-AHV Plugin - *[Nutanix-AHV Installation with BIM](#)*.

#### See also:

- Go back to *Nutanix-AHV Features*
- Go back to *Nutanix-AHV Backup and Restore Strategies*
- Go to *Nutanix-AHV Configuration*
- Go to *Nutanix-AHV Operations*
- Go to *Nutanix-AHV Troubleshooting*
- Go to *Nutanix-AHV Limitations*

Go back to the *main Nutanix-AHV Plugin page*.

**See also:**

Go back to:

- *Nutanix-AHV Features*

Go to:

- *Nutanix-AHV Installation*
- *Nutanix-AHV Configuration*
- *Nutanix-AHV Operations*
- *Nutanix-AHV Troubleshooting*
- *Nutanix-AHV Limitations*

Go back to the *main Nutanix-AHV Plugin page*.

## 4 Configuration

The following article presents the configuration of the Nutanix-AHV Plugin.

The plugin is configured using `Plugin Parameters` defined in the “Include” section of a FileSet resource (Bacula Director configuration).

### 4.1 General Parameters

The following Nutanix-AHV Plugin parameters impact any type of Job (Backup, Restore, Query).

**abort\_on\_error**[=`<0 or 1>`] Specifies whether or not the plugin should abort its execution if a fatal error happens during backup or restore. This parameter is optional. The default value is `0`.

**user**=`<String>` Specifies the user name used to access Nutanix-AHV API.

**password**=`<String>` Specifies the password used to access the Nutanix-AHV API. This parameter is optional and the password can be stored in a passfile as `host:user = password` tuple. An example of such a tuple would be `10.0.1.1:admin = password`. The passfile can be either at `/opt/bacula/snapmgr.conf` or at `/opt/bacula/etc/snapmgr.conf`. A `snapmgr.conf` file may have more than one tuple in it in the case that a Bacula FD is backing up more than one Nutanix environment.

**host**=`<String>` Specifies the location of the Nutanix-AHV API.

**port**=`<Number>` Specifies the port to access the Nutanix-AHV API located at host. This parameter is optional and if not specified its default value is `9440`.

**proxy\_vm**=`<String>` The name of the proxy VM used to run the Bacula File Daemon with the Nutanix-AHV plugin.

**working\_dir=<String>** Location of the working directory. This parameter is optional and if not specified its default value is `/opt/bacula/working`.

**debug=[0,9]** Specifies the level of debug with `0` being no debug and `9` being the highest level of debug. Warnings and errors are always sent to the joblog and if any debug level is set those messages are sent to the debug file as well. For the Nutanix-AHV plugin `1` displays debug level message, `2` displays trace level message. Any value higher than `2` displays additional information about external libraries that handle those values on their own. This parameter is optional. The default value is `0`

**See also:**

Go to:

- [Backup Parameters](#)
- [Restore Parameters](#)
- [Fileset Examples](#)
- [bconsoleQueryCommands](#)

Go back to the [main Nutanix-AHV Configuration page](#).

Go back to the [main Nutanix-AHV page](#).

## 4.2 Backup Parameters

---

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

---

**include=<Java Regexp>** Specifies a list of guest VM names to backup.

**exclude=<Java Regexp>** Specifies a list of guest VM names to not backup.

**vm=<guest VM name>** Specifies the name of a single guest VM to backup.

**application\_consistent=<true|try|false>** Specifies if the snapshot taken during backup should be application consistent `true`, crash consistent `false` or try application consistent first and fallback onto crash consistent if it fails `try`. This parameter is optional and its default value is `try`.

**dat\_file=<String>** Specifies the name of the `.dat` file used to keep track of dependencies between snapshot. This argument is optional and if not specified its default value is `nutanixAhv.dat`

The use of regular expressions in the parameters `include=` and `exclude=` must be a Java compatible regular expression.

In order to be backed up the guest VM must match the `include=...` predicate and not match the `exclude=...`. A guest VM that matches the `vm=...` will be backed up regardless of the include/exclude specifications.

By default all guest VMs match the include predicate and not the exclude. Therefore, if none of the parameters `vm=...`, `include=...` and `exclude=...` are provided, all available guest VMs hosted on the Nutanix-AHV hypervisor will be backed up.

On the other hand if the parameter `vm=...` is specified all guest VM will no longer match the include predicate. This means that if only `vm=...` parameter is specified no other guest VM will be backed up.

See [FileSet Examples](#) for examples of include/exclude/vm setups.

**See also:**

Go back to:

- [General Parameters](#)



Go to:

- [Restore Parameters](#)
- [Fileset Examples](#)
- [bconsoleQueryCommands](#)

Go back to the [main Nutanix-AHV Configuration page](#).

Go back to the [main Nutanix-AHV page](#).

### 4.3 Restore Parameters

**disconnect\_network**[=**<0 or 1>**] Specifies whether restored network interfaces need to be disconnected from their network. Default value is 0 (false)

**network\_address**=**<String>** Specified when a network interface should request a static IP address at restore. This argument is a single string of **name:ip\_address** tuples. See the FileSet examples.

**new\_hostname**=**<String>** Specified when a guest VM should be restored with a specific name.

**where**=**<String>** If this parameter is set, then the disks will be restored locally at the parameter value location. Each disk will be restored as **<where>/<uuid>.disk** file.

During restore the Nutanix-AHV plugin uses the generic plugin parameters to access the Nutanix-AHV API. The parameters may be modified if necessary at restore time.

A restore job can only restore one guest VM at a time. To select the relevant VM the user should interact with Bacula bconsole and mark all files in the guest VM folder identified by its UUID.

```
cwd is: /
$ ls
$ @nutanix-ahv/
$ cd @nutanix-ahv
cwd is: /@nutanix-ahv/
$ ls
$ 690d74f9-9ce4-45fb-9b23-149afebe18f7/
$ cd 690d74f9-9ce4-45fb-9b23-149afebe18f7/
cwd is: /@nutanix-ahv/690d74f9-9ce4-45fb-9b23-149afebe18f7/
$ ls
690d74f9-9ce4-45fb-9b23-149afebe18f7.1646238530898.conf
eef9d485-dba3-4e3f-b828-46a8c3412bec.1646238549482.abmp
eef9d485-dba3-4e3f-b828-46a8c3412bec.1646238549482.bmp
eef9d485-dba3-4e3f-b828-46a8c3412bec.1646238549493.bvmdk
$ cd ..
cwd is: /@nutanix-ahv/
$ mark 690d74f9-9ce4-45fb-9b23-149afebe18f7/*
4 files marked.
$ done
```

**See also:**

Go back to:

- [General Parameters](#)
- [Backup Parameters](#)

Go to:

- [Fileset Examples](#)
- [bconsoleQueryCommands](#)

Go back to the [main Nutanix-AHV Configuration page](#).

Go back to the [main Nutanix-AHV page](#).

## 4.4 Fileset Examples

In the example below, a single guest VM with a name of “VM1” will be backed up.

```
FileSet {
  Name= Nutanix_single
  Include {
    Plugin="nutanix-ahv: user=root host=1.2.3.4 proxy_vm=proxyVM vm=VM1"
  }
}
```

In the example below, all guest VMs which have prod in their name will be backed up.

```
FileSet {
  Name= Nutanix_prod
  Include {
    Plugin="nutanix-ahv: user=root host=1.2.3.4 proxy_vm=proxyVM include=(.*)prod(.*)"
  }
}
```

In the example below, all guest VMs which have prod in their name but do not start with test will be backed up.

```
FileSet {
  Name= Nutanix_no_test
  Include {
    Plugin="nutanix-ahv: user=root host=1.2.3.4 proxy_vm=proxyVM include=(.*)prod(.*)
    ↪) exclude=^test(.*)"
  }
}
```

In the example below, all prod VMs will be backed up. All test VMs will be ignored except for a VM named exception.test

```
FileSet {
  Name= Nutanix_prod_no_test_except
  Include {
    Plugin="nutanix-ahv: user=root host=1.2.3.4 proxy_vm=proxyVM include=(.*)prod(.*)
    ↪) exclude=(.*)test(.*) vm=exception.test"
  }
}
```

In the example below, at restore time the selected guest VM will have its network interfaces disconnected and the network interface linked to net1 will request the 10.0.110.11 ip address.

```
FileSet {
  Name= Nutanix_disconnect
```

(continues on next page)

```

Include {
    Plugin="nutanix-ahv: user=root host=1.2.3.4 proxy_vm=proxyVM disconnect_network_
↪network_address=net1:10.0.110.11"
}
}

```

**Note:** There are different ways to use `disconnect_network` and `abort_on_error`: `<parameter>=1` is strictly equal to the presence of `<parameter>` inside the plugin line. In the same spirit the absence of such parameter in the plugin line is equivalent to `<parameter>=0`.

In the example below, at restore time the selected guest VM will have its network interface connected to `net1` request the ip `10.0.110.11` and the network interface `net2` request the address `127.0.0.11`. Note that the `disconnect_network network_address` must be present in the FileSet resource at the backup time to benefit from it at restore time.

```

FileSet {
    Name= Nutanix_net
    Include {
        Plugin="nutanix-ahv: user=root host=1.2.3.4 proxy_vm=proxyVM network_
↪address=net1:10.0.110.11,net2:127.0.0.11"
    }
}

```

**See also:**

Go back to:

- [General Parameters](#)
- [Backup Parameters](#)
- [Restore Parameters](#)

Go to:

- [bconsoleQueryCommands](#)

Go back to the [main Nutanix-AHV Configuration page](#).

Go back to the [main Nutanix-AHV page](#).

## 4.5 bconsole Query Commands

The Bacula Enterprise Nutanix-AHV Plugin supports also the query parameter. Bacula queries the Nutanix-AHV API and receives answers in the form of `key=value`. The Nutanix-AHV Plugin supports three query parameters. When none of them is specified, the message: `Query not recognized possible value = {CONNECTION, MACHINES, NETWORK}` is displayed.

## Connection

This query sends a REST API request to check whether the current plugin parameters allow connections to the Nutanix-AHV server. The example below shows the execution of a correctly formatted CONNECTION query and response.

When the query parameter is CONNECTION, the returned value will be either OK or NOK, indicating whether the connection was successful or not.

```
.query plugin="nutanix-ahv: user=admin host=1.2.3.4" client=client-fd_
↔parameter=CONNECTION
info=Query
CONNECTION=OK
```

## VM

This query sends a REST API request that lists all available guest VMs. This query displays one key=value per virtual machine where key is VM and value is the name of the virtual machine.

The example below shows the execution of a correctly formatted VM query that finds three different guest VMs.

```
.query plugin="nutanix-ahv: user=admin host=1.2.3.4" client=client-fd parameter=VM
info=Query
VM=Virtual1
VM=Virtual2
VM=Virtual3
```

## Network

This query sends a REST API request that lists all available networks. This query displays one key=value tuple per network available where key is NETWORK and value is the network's name.

The example below show values output by a correctly formatted NETWORK query that finds two networks.

```
.query plugin="nutanix-ahv: user=admin host=1.2.3.4" client=backuparnaud-fd_
↔parameter=NETWORK
info=Query
NETWORK=Managed
NETWORK=Vlan-Nutanix
```

### See also:

Go back to:

- [General Parameters](#)
- [Backup Parameters](#)
- [Restore Parameters](#)
- [Fileset Examples](#)

Go back to the [main Nutanix-AHV Configuration page](#).

Go back to the [main Nutanix-AHV page](#).

### See also:

Go back to:

- [Nutanix-AHV Features](#)
- [Nutanix-AHV Backup and Restore Strategies](#)
- [Nutanix-AHV Installation](#)

Go to

- [Nutanix-AHV Operations](#)
- [Nutanix-AHV Troubleshooting](#)
- [Nutanix-AHV Limitations](#)

Go back to the [main Nutanix-AHV Plugin page](#).

## 5 Operations

The following article describes details regarding backup, ingestion, restore and network restore with Bacula Enterprise Nutanix-AHV Plugin.

---

**Note:** The Nutanix-AHV plugin allows the request of static IP addresses only on a managed network.

---

### 5.1 Backup

The backup of a single guest VM takes the following steps:

- Take a snapshot of the guest VM through a hidden Nutanix API endpoint.
- Export guest VM metadata configuration for future restore.
- Backup every disks on the guest VM by ingestion on proxy VM.
- Add new a snapshot to the tracker and look for snapshots to delete.

**See also:**

Go to:

- [Restore](#)
- [Interactive Snapshot Deletion](#)

Go back to the [main Operations page](#).

Go back to the [main Nutanix-AHV page](#).

### 5.2 Restore

The Nutanix-AHV plugin restores data as a new guest VM.

The restore is analog to the backup process and goes through the following steps for each successive snapshot of the guest VM to be restored.

- Receive configuration metadata file.
- Receive two lists: one for the changed regions and another for zeroed regions.
- Receive the raw disk data file(s).

- Add a new virtual disk on the SCSI bus of the proxy VM (hotadd).
- Write all data regions back onto the disk(s).
- Once all disks have been restored create a new guest VM from the metadata configuration file
- Detach all restored disks from the proxy VM.
- Move all restored disks from proxy VM to the freshly restored guest VM.

**See also:**

Go back to:

- [Backup](#)

Go to:

- [Interactive Snapshot Deletion](#)

Go back to the [main Operations page](#).

Go back to the [main Nutanix-AHV page](#).

### 5.3 Interactive Snapshot Deletion

The Bacula Enterprise Nutanix-AHV Plugin backend comes with an additional feature to manually keep track of snapshots, which is a simple command line tool located in `/opt/bacula/bin`. To launch the CLI the user just needs to execute the backend script with an additional `delete` parameter. The script asks the user for their username and the Nutanix-AHV host. The password will be read from the `snapmgr.conf` file.

The example below shows a regular execution of the interactive snapshot deletion feature.

```

root@backupvm:~# /opt/bacula/bin/nutanix_ahv_backend delete
Enter credential   USER=
admin
Enter credential   HOST=
10.0.100.90
Do you want to treat virtual machine
    VM_NAME=TestTest
    VM_UUID=e95339ab-a15a-472e-977a-961e7f5440b1
[y]es / [n]o
y
Start interactive delete for UUID=e95339ab-a15a-472e-977a-961e7f5440b1
Snapshots detected= 11
Do you want to delete snapshot n°1
    UUID=0929948b-04ee-4df7-b7d2-1b451be9ad06
    SNAPSHOT_NAME=NutanixAhv.2022-02-15_11.07.41_09
[y]es / [n]o

```

**See also:**

Go back to:

- [Backup](#)
- [Restore](#)

Go back to the [main Operations page](#).

Go back to the [main Nutanix-AHV page](#).

**See also:**

Go back to:

- [Nutanix-AHV Features](#)
- [Nutanix-AHV Backup and Restore Strategies](#)
- [Nutanix-AHV Installation](#)
- [Nutanix-AHV Configuration](#)

Go to:

- [Nutanix-AHV Troubleshooting](#)
- [Nutanix-AHV Limitations](#)

Go back to the [main Nutanix-AHV Plugin page](#).

## 6 Troubleshooting

This article presents troubleshooting for the Nutanix-AHV Plugin.

If the proxy VM boots to an initramfs shell after a failed backup/restore it is probably due to the fact that some of the disks that were mounted on the proxy VM are still attached.

To solve this issue the user has to manually remove the supernumerary disks manually through the PRISM console.

Another way to remove the disk is to issue REST API commands. A simple bash script like the following example will do the trick:

---

**Note:** This script is pretty aggressive and will remove all disks from index 1 to 10 without asking anything. It should be used carefully and adapted if the initial proxy VM configuration has more than one disk on the SCSI bus.

---

```
for i in {1..10}
do
curl -k -X POST --header 'Content-Type: application/json' --header 'Accept: application/
↪ json' -d '{
  "uuid": "$proxyVmUuid",
  "vm_disks": [
    {
      "disk_address": {
        "device_bus": "SCSI",
        "device_index": '$i'
      },
      "is_cdrom": false
    }
  ]
}' 'https://$HOST:$PORT/PrismGateway/services/rest/v2.0/vms/$proxyVmUUID/disks/detach' -
↪ u '$USER:$PASSWORD'
done
```

The script will try to unmount all disks from index 1 up to 10. Index 0 is skipped since the proxy VM first disk is the one containing the VM data.

**See also:**

Go back to:

- [Nutanix-AHV Features](#)
- [Nutanix-AHV Backup and Restore Strategies](#)
- [Nutanix-AHV Installation](#)
- [Nutanix-AHV Configuration](#)
- [Nutanix-AHV Operations](#)

Go to:

- [Nutanix-AHV Limitations](#)

Go back to the [main Nutanix-AHV Plugin page](#).

## 7 Limitations

This article presents limitations of the Nutanix-AHV Plugin.

- The proxy VM must be a VM in the Nutanix-AHV environment running a Linux distribution for the plugin to work.
- Disks located on the IDE bus cannot be backed up. Disk tray on the other hand will be backed up.
- Content mounted on a virtual disk tray will not be restored.
- Only one VM may be restored at a time. If a backup job contains more than one VM, then each VM will need to be restored separately.
- Virtual full jobs are not supported.

**See also:**

Go back to:

- [Nutanix-AHV Features](#)
- [Nutanix-AHV Backup and Restore Strategies](#)
- [Nutanix-AHV Installation](#)
- [Nutanix-AHV Configuration](#)
- [Nutanix-AHV Operations](#)
- [Nutanix-AHV Troubleshooting](#)

Go back to the [main Nutanix-AHV Plugin page](#).

Go back to the main [Dedicated Backup Solutions page](#).