



Backup and Restore of Avi Vantage Configuration

Avi Technical Reference (v20.1)

Copyright © 2021

Backup and Restore of Avi Vantage Configuration

[view online](#)

Periodic backup of the Avi Vantage configuration database is recommended. This database defines all clouds, all virtual services, all users, and so on. Any user capable of logging into the admin tenant is authorized to perform a backup of the entire configuration, i.e., of all tenants. A restore operation spans all the same entities, but can only be performed by the administrator(s) capable of logging into one of the Controllers using SSH or SCP.

It is a best practice to store backups in a safe, external location, in the unlikely event that a disaster destroys the entire Avi Controller (or cluster), with no possibility of remediation. Based on how often the configuration changes, a recommended backup schedule could be daily or even hourly.

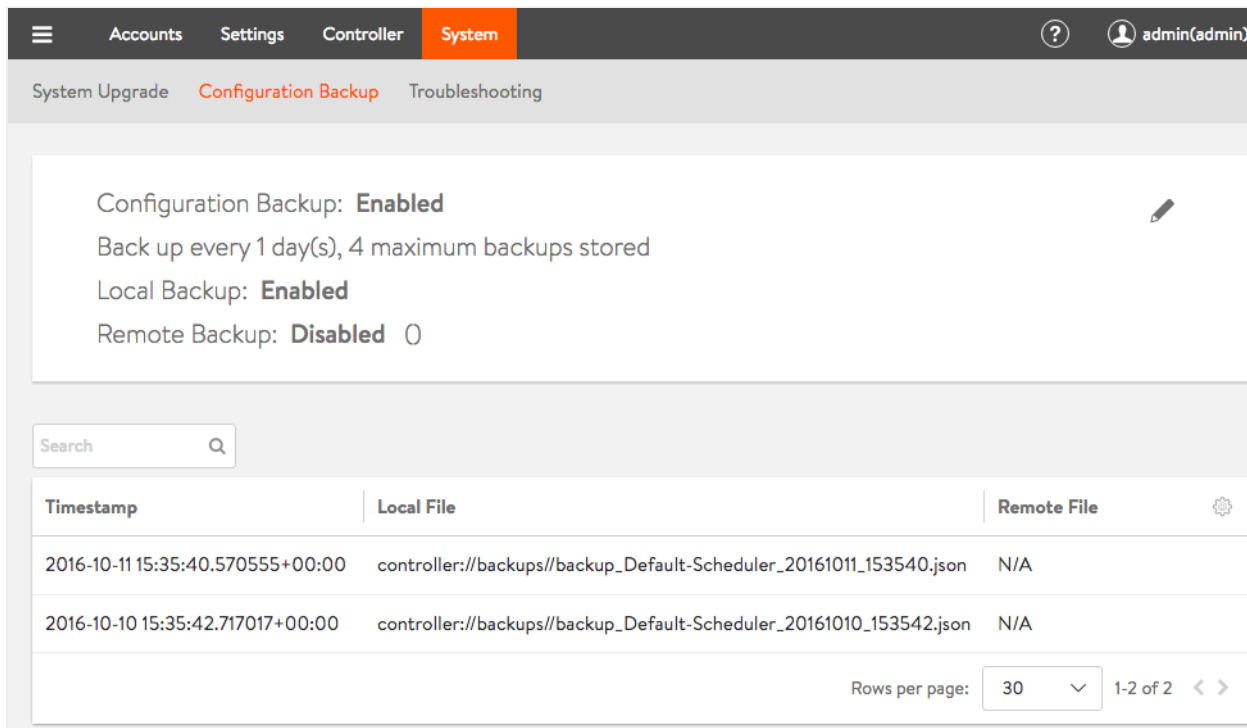
Backing Up the Avi Vantage Configuration

To back up the Avi Vantage configuration, use the Avi UI, CLI commands or API commands shown in this section. Backups can be scheduled or on-demand.

Scheduled Backup via Avi UI

To view or edit the configuration backup scheduler's current settings, an admin-tenant user first navigates to Administration > System > Configuration Backup as shown below.

Note: The scheduled backups get stored in `/var/lib/avi/backups/` on all Avi Controllers in the cluster.



To effect changes, click on the edit option as shown in the above screenshot. The Backup Scheduler editor appears as shown below:

Edit Backup Scheduler: Default-Scheduler ✕

General

Enable Configuration Backup

Periodic Backup

Frequency [?] Frequency Unit [?]

Backup Passphrase [?]

Backup Destination

Local (on Controller) [?]

File Prefix [?] Number of backups to store [?]

Remote Server [?]

Save

- **Enable Configuration Backup:** Turns scheduled backups on or off.
- **Frequency:** A value from 0 to 60 to combine with Frequency Unit to determine how often backups are to be taken. 0 indicates the backup sequence has no end time.
- **Frequency Unit:** Backups occur daily by default. Use this field to change the units to minutes, hours, weeks or months.
- **Backup Passphrase:** A passphrase is used to encrypt all sensitive fields contained within the backup. Choose a phrase that is not easy to guess and guard it carefully. Data cannot be restored without it. Starting with Avi Vantage release 20.1.1, if a passphrase is not provided, the default backup passphrase is accepted as the export/import passphrase.
- **Local (on Controller):** On by default, this switch tells Avi Vantage to preserve the number of indicated backups on the Controller. One can choose the local and remote backup options independently.
- **File Prefix:** An optional character string that permits customers to customize the backup archive filename.
- **Number of backups to store:** A number ranging from 0 to 20, default is 4. 0 is equivalent to unchecking the Local option. The oldest backup is deleted after the most recent backup successfully completes.
- **Remote Server:** This option is off by default. It is recommended that a remote destination be specified in case the Avi Controller cluster fails in a non-recoverable fashion. One can choose the local and remote backup options independently. Turning this option on causes the Controller to log onto the indicated server using SSH user credentials, and then secure copy (*scp*) the backup data to the indicated directory.
 - **Server Address** is an FQDN or IP address reachable from the Controller.
 - **User Credentials:** Use the pulldown menu to select from a previously-defined SSH user or accept the option to create one.

- **Directory:** If a target directory other than the SSH user's home directory is desired, specify an absolute or relative pathname pointing to a directory to which the SSH user has write access.

Scheduled Backup via CLI

```
[admin:10-10-24-52]: > configure scheduler Default-Scheduler
+-----+
| Field          | Value                               |
+-----+
| uuid           | scheduler-b5f7e673-8818-44d1-8f74-45238cc08235 |
| name           | Default-Scheduler                   |
| enabled        | True                                 |
| run_mode       | RUN_MODE_PERIODIC                   |
| start_date_time | 2016-10-09T15:35:46.220623          |
| frequency      | 1                                    |
| frequency_unit | SCHEDULER_FREQUENCY_UNIT_DAY        |
| backup_config_ref | Backup-Configuration                |
| scheduler_action | SCHEDULER_ACTION_BACKUP             |
| tenant_ref     | admin                                |
+-----+

[admin:10-10-24-52]: configure backupconfiguration Backup-Configuration
+-----+
| Field          | Value                               |
+-----+
| uuid           | backupconfiguration-5d65f12e-5da1-49e0-b703-ec65ae9a39c6 |
| name           | Backup-Configuration                |
| save_local     | True                                 |
| maximum_backups_stored | 4                                    |
| tenant_ref     | admin                                |
+-----+
```

One can specify the value of `start_date_time` from the CLI (not possible via the Avi UI):

```
[admin:10-10-24-52]: > configure scheduler Default-Scheduler
[admin:10-10-24-52]: scheduler> no enabled
[admin:10-10-24-52]: scheduler> start_date_time 2017-05-11T00:00:00
Overwriting the previously entered value for start_date_time
[admin:10-10-24-52]: scheduler> save
[admin:10-10-24-52]: > configure scheduler Default-Scheduler
[admin:10-10-24-52]: scheduler> enabled
Overwriting the previously entered value for enabled
[admin:10-10-24-52]: scheduler> save
```

Scheduled Backup via API

In this example a PUT changes the scheduler frequency to 1 week:

```
PUT : api/scheduler/
{'_last_modified': u'1476209663670990',
'backup_config_ref': 'https://10.10.24.52/api/backupconfiguration/backupconfiguration-5d65f12e-5da1-49e0-b703-ec65ae9a3
'enabled': True,
'frequency': 1,
'frequency_unit': u'SCHEDULER_FREQUENCY_UNIT_WEEK',
'name': u'Default-Scheduler',
'run_mode': u'RUN_MODE_PERIODIC',
'scheduler_action': u'SCHEDULER_ACTION_BACKUP',
'start_date_time': u'2016-10-09T15:35:46.220623',
'tenant_ref': u'https://10.10.24.52/api/tenant/admin',
'url': 'https://10.10.24.52/api/scheduler/scheduler-b5f7e673-8818-44d1-8f74-45238cc08235',
'uuid': u'scheduler-b5f7e673-8818-44d1-8f74-45238cc08235'}
```

On-demand Backup via Avi CLI

To back up the Avi Vantage configuration on-demand, at any arbitrary time, use the following CLI command:

```
> export configuration file /tmp/avi_config.json full_system
Please enter the passphrase to encrypt configuration:
Downloaded the attachment to /tmp/avi_config.json
Completed writing the export configuration to /tmp/avi_config.json
```

On-demand Backup via REST API

To back up the Avi Vantage configuration on-demand, at any arbitrary time, use the following API request:

```
GET https://[CONTROLLER-IP]/api/configuration/export?full_system=true
```

To also include a passphrase, use one of the following options:

In versions 16.1.3 and above:

```
GET https://[CONTROLLER-IP]/api/configuration/export?full_system=true&passphrase=[PASSPHRASE]
```

In versions 16.2 and above, use the following POST method and include passphrase in the JSON data:

```
POST https://[CONTROLLER-IP]/api/configuration/export?full_system=true
JSON data: {"passphrase": "[PASSPHRASE]"}
```

Make sure to replace [CONTROLLER-IP] with the IP address of the Avi Controller (if using a single Avi Controller node), or the IP address of the Avi Controller cluster.

On-demand Backup Script Utilizing Python

Refer to https://github.com/avinetworks/sdk/blob/master/python/avi/sdk/samples/avi_config_backup.py

Configuring Backup using Amazon S3

Starting with Avi Vantage release 18.2.3, backup configuration can be stored on Amazon S3 bucket as well.

To enable backup configuration on Amazon S3, use the `configure backupconfiguration` command and set value of the `upload_to_s3` flag to true.

Provide the value of the following attributes to save the backup file on the Amazon S3 bucket for the required instance.

- `aws_access_key`: Access Key ID
- `aws_secret_access`: Secret Access Key
- `aws_bucket_id`: Name of the S3 bucket

```
[admin:10-1-1-1]: configure backupconfiguration Backup-Configuration
```

Note:

For enabling Avi Controller's backup, you should have write permission to S3 bucket.

For the detailed information on the Access Key ID and Secret Access Key, refer to [AWS Cross-Account AssumeRole Support](#).

Restoring the Avi Vantage Configuration

If the unlikely should occur and a disaster completely destroys the Avi Controller (or entire cluster), the device/VM hosting the Avi Controller(s) should first be restored to factory default using `flushdb.sh`. Failure to do so can prevent the Controller from coming up. If there is a prev partition (rename or delete the prev partition). The prev partition can either be `root1` or `root2`
`mv root1/root2 prev_back`

Steps to check the Partition Mapping are as listed below:

1. `sudo cat /proc/cmdline`

You can observe an output with either `root1` or `root2` as a current partition as shown below:

For example, we see `root1` below as current partition:Output

```
root=UUID=f4a947e1-7efb-4345-9eac-1ff680fc50e0 subroot=/root1 net.ifnames=0 biosdevname=0 console=tty0 console=
```

2. Go to `/host` directory and rename the prev partition as shown below:

```
cd /host
ls -lrth <- This is to see if you have root1 and root2 directories.
mv root2 prev_bak ----> as root2 is prev partition
```

Thereafter, the following script can be used to automate the configuration recovery process:

```
<code> /pt/avi/scripts/restore_config.py</code>
```

Note: Starting with Avi Vantage version 18.2, prev partition needs to be removed.

This script imports the backup configuration onto the Avi Controller. If restoring an Avi Controller cluster, this script restores the configuration and also re-adds the other two nodes to the cluster. 1. Create three new Avi Controllers with the same IP address as the original cluster members. (Avi Vantage currently supports only static IP addresses.) At this point, other than having an IP address, each Avi Controller node should be in its factory default state. 2. Log onto one of the Avi Controller nodes using SSH or SCP. Use the default administrator credentials. 3. Run the restore command or script: * Copy backup file via SCP:

```
scp /var/backup/avi_config.json admin@<controller-ip>://tmp/avi_config.json
```

- Run restore command locally via SSH:

```
/opt/avi/scripts/restore_config.py --config CONFIG --passphrase PASSPHRASE --do_not_form_cluster DO_NOT_FORM_CL
```

In the above command line:

- **CONFIG** is the path of the configuration file.
- **PASSPHRASE** is the export configuration passphrase.
- **DO_NOT_FORM_CLUSTER** causes cluster formation to be skipped.
- **VIP** is the virtual IP address of the Avi Controller.
- **FOLLOWER_IP [FOLLOWER_IP ...]** is a list of the IP addresses of the followers.
- **CLUSTER_UUID** is the old cluster UUID to be restored.