



Flexible Upgrades for Avi Vantage

Avi Technical Reference (v20.1)

Copyright © 2021

Flexible Upgrades for Avi Vantage

[view online](#)

This article is only applicable for Avi Vantage release 18.2.6 and later versions. For Avi Vantage release prior to 18.2.6, refer to [Upgrading Avi Vantage Releases Prior to 18.2.6](#).

Overview

Starting with release 18.2.6, Avi Vantage supports improved and more flexible methods for upgrading the Avi Vantage system.

Note: Starting with Avi Vantage release 20.1.1 as per the HTTP/2 RFC, the cipher suites with TLS 1.2 listed [here](#) are not supported. Remove the ciphers before initiating upgrade to Avi Vantage version 20.1.1.

The following are the additional features for the Flexible Upgrades: * The upgrade is possible per SE group. The transition of all the SE groups to the new version may occur over a long period. * Upgrades of different SE groups are supported with different patch versions. * Rollback to the previous versions of Avi Vantage is non-disruptive.

From Avi Vantages prior to 18.2.6, the only available option is system-level (Avi Controller and SE Groups) upgrade. With Flexible Upgrades, the following options are available:

```
<th> Upgrades </th>
<th> Patch Ugrades</th>
<th> Rollback</th>
<th> Rollback Patch </th>
```

```
<td> System (Avi Controller and SE Groups)</td>
<td> System (Avi Controller and SE Groups</td>
<td> System (Avi Controller and SE Groups</td>
<td> System (Avi Controller and SE Groups</td>
```

```
<tr>
<td> Avi Controller only</td>
<td> Avi Controller only</td>
<td> Avi Controller only</td>
<td> Avi Controller only</td>
</tr>
<tr>
<td> Some or all the SE groups</td>
</tr>
```

Use Cases

- Scenarios when it is not possible to upgrade all SE groups to the newer version at the same time due to various business reasons such as logistics, confidence in the new software, etc.
- The configuration is blocked during the entire duration of the Avi Controller and SE upgrade. This is not acceptable in many deployments. With the new upgrade feature, the process is flexible and can be performed per SE group basis. The configuration is blocked for the entire duration if a system upgrade is performed till all Service Engines are upgraded.
- Using SE groups for data plane separation. Based upon the SE group segmentation, the upgrade is performed based upon the following attributes.
 - Application or product offering
 - Tenant
 - Production, pre-production and development environments
 - Cloud or environment (AWS, VMware, etc.)
 - Provide patches to only applications or SE groups that need them
 - Flexible scheduling
 - Self-service upgrades

Image Management and Service

Image service is the first step in the flexible upgrade work-flow. It is used to upload the image after which an upgrade operation can be initiated. The Avi Controller hosts images of different versions since SE groups could be potentially in different versions.

The Avi Controller should have additional disk space to host these images.

Avi Controller images for the major versions include the followings: * *controller.pkg* (for VM-based Avi Controller) * *controller_docker.tgz* (For Docker-based Controller)

Images for the patches include the followings: * *avi_patch.pkg* ? Full package * *controller_patch.pkg* ? Avi Controller package * *se_patch.pkg* ? SE patch package

As a part of the upload process, image service extracts files, metadata from the package. This information is not only presented to the user but also used in the upgrade process.

Notes: * Images from Avi Vantage release 17.2.8 onwards are upgradeable to an image for Avi Vantage release 18.2.6. The image prior to the release 17.2.8, should be migrated to 17.2.8 image after which it can be upgraded to 18.2.6. * Image service provides an ability to upload, query and delete Avi image(s) to the system. * Image service supports the upload of Avi patch packages. * Image upload can happen only on the cluster leader. It is not allowed from a cluster member.

Image Bundling

Avi Vantage now supports the composite image or the image bundle. The composite image of Avi Vantage consists of the followings: * Base image ? Controller image (*controller_docker.tgz*, *controller.pkg*, *controller ova*, *controller.qcow2*, etc) * Controller package ? It is an optional package * SE patch image ? It is an optional package

The upgrade workflow using the image bundle, or the composite image is the same as using the standard image. When the image bundle is used for upgrade, a patch image can also be applied along with the base image.

Uploading Image Using Avi CLI

The CLI for Avi Vantage release 18.2.6 provides better control of the upgrade operations leading to a consistent and predictable workflow.

For uploading the package use the `upload image filename <path-of-the-package>` command as shown below.

```
[admin:controller]: > upload image filename /tmp/controller.pkg
```

The following show command returns the details of the image metadata.

```
show image <image-name>
```

```
[admin:-controller]: > show image
+-----+-----+-----+-----+
| Name                               | UUID                               | Status           |
+-----+-----+-----+-----+
| 18.2.7-5000-20191009.205501 | image-fxxxx22-0f40-45de-8551-15xxxxxxx1fe | SYSERR_SUCCESS |
+-----+-----+-----+-----+
```

The existing API endpoints (prior to 18.2.6) are not supported. To know more about differences in CLI commands and APIs refer to [Comparison Table for Differences in CLIs Commands and APIs](#).

Uploading Image Service using Avi REST API

A POST operation is used to do an image upload. To get the image details in response, execute a GET API request.

- Use the following REST API to upload image for *controller.pkg*.
URI: /api/image
Method: POST

```
root@admin:-controller# curl -X POST -k https://10.58.3.27/api/image -u "admin:admin" -F file=@controller.pkg
```

- Use the following REST API to upload image for *controller_patch.pkg*.

```
root@admin:-controller-18.2.5-2p3-9002# curl -X POST -k https://10.58.3.27/api/image -u "admin:admin" -F file=@se_p
```

- Use the following API to delete the image provided, if it is not in use.
delete image <image-name>

Must-Checks for Upgrade

Prior to upgrade operations, various must-checks are executed to check the various mandatory and optional requirements for upgrade. The outputs message is exhibited as *error* message or as *Warning* message. *Warnings* can be skipped while 'Errors' cannot be over-ridden. API/CLI provides the *skip_warnings* option to control the above behavior.

For Avi CLI? This is directly integrated into the normal work-flow and there is no separate command.
For the REST API ? Add /preview/ at the end of APIs to get previews for that particular flow.

Upgrading Avi System (Avi Controller and SE Groups)

The configuration and placement of virtual services are blocked if it is a system-level upgrade till all the Service Engines are upgraded. Once these operations are completed, configuration on Avi Controller (except the configuration of virtual service and VIP) is allowed, irrespective of the SE group upgrade status.

Note: When upgrading from Avi Vantage versions 17.x or version lesser than 18.2.6 to Avi Vantage 20.1 and higher, in the Avi Controller, change the `DefaultTimeoutStartSec` (File: `/etc/systemd/system.conf`) to 120 seconds to avoid timeout during upgrade.

Using Avi CLI

Notes: * The auto-suggest option in the Avi CLI provides available values on pressing tab on your keyboard. * `skip_warnings` ? Use this option to skip any warnings and optional must checks.

The following are the various options available for Avi system upgrade.

- Use the `upgrade system image_ref <image name>` command to upgrade the system to a base image.

```
[admin:-controller]: >upgrade system image_ref 18.2.6-9000-20191031.063017
```

- Use the following to upgrade the system to a base image and a controller patch.

```
[admin:-controller]: >upgrade system image_ref 18.2.6-9134-20191101.042535 controller_patch_ref 18.2.6-9134-2p1
```

- Use the following to upgrade the system to a base image and an SE patch.

```
[admin:-controller]: >upgrade system image_ref 18.2.6-9134-20191101.042535 se_patch_ref 18.2.6-9134-2p1-2019080
```

- Use the following to upgrade the system to a base image, an Avi Controller patch, and an SE patch

```
[admin:-controller]: >upgrade system image_ref 18.2.6-9134-20191101.042535 controller_patch_ref 18.2.6-9134-2p1
```

Using Avi REST API

Image UUID can be obtained by Use the `GET /api/image` to obtain image UUID.

The following are the various REST API options available for Avi system upgrade.

- Use the following API to upgrade the system to a base image.

API: `/api/upgrade`

Method: `POST`

JSON Data:

```
{
  'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4',
  'system': true
}
```

- Use the following API to upgrade the system to a base image and a controller patch.

API: /api/upgrade

Method: POST

JSON Data:

```
{
  'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4',
  'controller_patch_uuid': 'image-e3aaad68-5aaf-485a-8bd9-1db3ec562d6a',
  'system': true
}
```

- Use the following API to upgrade the system to a base image and an SE patch.

API: /api/upgrade

Method: POST

JSON Data:

```
{
  'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4',
  'system': true,
  'se_patch_uuid': 'image-e3aaad68-5aaf-485a-8bd9-1db3ec562d6a',
  'skip_warnings': True
}
```

- Use the following API to upgrade the system to a base image, an Avi Controller patch, and an SE patch

API: /api/upgrade

Method: POST

JSON Data:

```
{
  'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4',
  'controller_patch_uuid': 'image-e3aaad68-5aaf-485a-8bd9-1db3ec562d6a',
  'system': true,
  'se_patch_uuid': 'image-e88aaad68-5aaf-485a-8bd9-1db3ec562d6a'
}
```

Upgrading Avi Controller

Using Avi CLI

Login to the Avi shell prompt and use the following upgrade commands for various options.

- Use the `upgrade controller image_ref <image name>` command to upgrade the Avi Controller to a base image.

```
[admin:-controller]: >upgrade controller image_ref 18.2.6-9000-20191031.063017
```

- Use the `upgrade controller image_ref <image name>controller_patch_ref <patch name>` command to upgrade the Avi Controller to a base image and an Avi Controller patch.

```
[admin:-controller]: >upgrade controller image_ref 18.2.6-9134-20191101.042535 controller_patch_ref 18.2.6-9134
```

Using Avi REST API

- Use the following API to upgrade the Avi Controller to a base image.

API: `/api/upgrade`

Method: POST

JSON Data:

```
{
  'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4'
}
```

- Use the following API to upgrade an Avi Controller to a base image and an Avi Controller patch.

API: `/api/upgrade`

Method: POST

JSON Data:

```
{
  'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4',
  'controller_patch_uuid': 'image-e3aaad68-5aaf-485a-8bd9-1db3ec562d6a'
}
```

Upgrading SE Group

This interface is used to upgrade all or some of the SE groups.

Using Avi CLI

Login to the Avi shell prompt to use the various options available for SE group update.

- Use the `upgrade segroup se_group_refs Default-Group image_ref<image name>` command to upgrade an SE group to the Controller image.

```
[admin:-controller]: >upgrade segroup se_group_refs Default-Group image_ref 18.2.6-9134-20191101.042535
```

- Use the `upgrade segroup se_group_refs Default-Group image_ref <Controller image> se_patch_ref <SE patch name>` command to upgrade an SE group to the Controller image and the SE patch image.

```
[admin:-controller]: >upgrade segroup se_group_refs Default-Group image_ref 18.2.6-9134-20191101.042535 se_patch
```

Using Avi REST API

SE Group UUID can be obtained by the `GET /api/serviceenginegroup` API.

The followings are the additional options for SE group upgrade:

- **Disruptive ?** This is used to disable non-disruptive mechanism to facilitate a faster upgrade. If enabled, the SE(s) are upgraded in a disruptive manner. The default value is false.
- **Suspend-on-failure ?** This option suspends the upgrade of subsequent SE(s) within a SE-group when a failure is encountered in the SE upgrade path. The default value is false.

The followings are the different APIs for the SE group upgrade: * Use the following API to upgrade the SE group to the Controller image.

API: `/api/upgrade`

Method: POST

JSON Data:

```
{
  'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4',
  'se_group_uuids': [
    'serviceenginegroup-e553b1a6-4851-4e82-ad12-cecc4bbda6c7'
  ]
}
```

- Use the following with the additional SE Group options ? Disruptive and Suspend_on_failure.

API: `/api/upgrade`

Method: POST

JSON Data:

```
{
  'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4',
  'se_group_uuids': [
    'serviceenginegroup-e553b1a6-4851-4e82-ad12-cecc4bbda6c7'
  ],
  'disruptive': true,
  'suspend_on_failure': true
}
```

- Use the following API to upgrade the SE group to the Controller image and the SE patch image.

API: `/api/upgrade`

Method: POST

JSON Data:

```
{
  'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4',
  'se_patch_uuid': 'image-e3aaad68-5aaf-485a-8bd9-1db3ec562d6a',
  'se_group_uuids': [
    'serviceenginegroup-e553b1a6-4851-4e82-ad12-cecc4bbda6c7'
  ]
}
```

```

    ]
}

```

Additional Options for SE Group Upgrade

The following upgrade options are available for upgrading SE group.

Option	Behaviour	Notes
SUSPEND_UPGRADE_OPS_ON_FAILURE	This option is used to suspend the upgrade-operations (Upgrade/Patch) on SE-Group if the SE(s) hit an issue and does NOT come up during the upgrade operations.	It is enabled by default. This option serializes the SE upgrades in the SE group upgrade. It increases the overall upgrade time for the entire SE group. Batch size is used to decrease the upgrade time. Even if the SEs does not have scaled-out virtual services, it still upgrades serially.
CONTINUE_UPGRADE_OPS_ON_FAILURE	This option is used to continue the upgrade or patch upgrade operations on SE group even if the SE(s) hit an issue and does not come up during the upgrade operations. Service disruption can be observed.	This option parallelizes the SE upgrade in the SE group upgrade if SEs does not have scaled-out virtual services. If SEs have scaled-out virtual services, then it continue with serial upgrades.
Disruptive	This option is used to disable the non-disruptive nature of SE upgrade. It is used to upgrade all the SE(s) in the group to the next version irrespective of the traffic disruption.	This option is disabled by default. All SE(s) will be upgraded in parallel, irrespective of scaled out virtual service existence. Traffic/Service disruption will take place.

Upgrading using Patch Release

The followings are the available options for patch upgrade: * System ? Patch upgrade for Avi Controller and all SE groups * Controller ? Patch upgrade for the Avi Controller alone. * SE group ? Patch upgrade for some or all the SE groups.

Notes:

The following are a few points for a patch upgrade process: * An image along with a patch can be applied. * The image and the patch must have the same base version. * A patch cannot be applied without applying the image.

- Compatibility checks prevent incorrect patches from getting applied to different versions.

To upload the image for patch upgrades, refer to

Patch Upgrade for Avi System

Use the following CLI command for the base image upgrade with a patch image.

```


```

```
[admin:controller]: > upgrade system image <image-name> controller_patch <controller-patch-name> se_patch <se-patch-name>
[admin:controller]: > upgrade system image 18.2.6 controller_patch 18.2.6-1p1 se_patch 18.2.6-1p1
```

2. Use the `upgrade system image_ref <image name> controller_patch_ref <SE patch name>` command for an Avi Vantage system upgrade with a Controller patch.

```
[admin:-controller]: upgrade system image_ref 18.2.6-9000-20191031.063017 controller_patch_ref 18.2.6-2p1-20191031.063017
```

3. Use the `upgrade system image_ref <image name> se_patch_ref <SE patch name>` command for an Avi Vantage system upgrade with only SE patch.

```
[admin:-controller]: upgrade system image_ref 18.2.6-9000-20191031.063017 se_patch_ref 18.2.6-2p1-20191031.063017
```

4. Use the `upgrade system image_ref <image name> controller_patch_ref <Controller patch image> se_patch_ref <SE patch image>` command for the system upgrade with both Controller and SE patch.

```
[admin:-controller]: upgrade system image_ref 18.2.6-9000-20191031.063017 controller_patch_ref 18.2.6-2p1-20191031.063017 se_patch_ref 18.2.6-2p1-20191031.063017
```

Patch Upgrade for Avi Controller

This interface is used to patch upgrade for the Avi Controller.

Using Avi CLI

Use the `upgrade controller image_ref <image name> controller_patch_ref <Controller patch image>` command to upgrade the Avi Controller with a patch.

```
[admin:-controller]: upgrade controller image_ref 18.2.6-9000-20191031.063017 controller_patch_ref 18.2.6-2p1-20191031.063017
```

```
[admin:controller]: > patch controller <patch-name>
```

```
[admin:controller]: > patch controller controller_patch 18.2.5-5p1
```

Using Avi REST API

```
POST api/upgrade JSON data:{?controller_patch_uid?: <image-uid>}
```

Patch Upgrade for SE Group

SE groups can be of different versions and different versions of patch can be applied. Use the `upgrade segroup image_ref <image name> se_group_refs Default-Group se_patch_ref <patch for the SE Group>` command to upgrade specific SE groups along with a patch.

```
[admin:-controller]: upgrade segroup image_ref 18.2.6-9000-20191031.063017 se_group_refs Default-Group se_patch_ref 18.
```

Note: Patch name and patch uuid is retrieved from the image service.

Rollback

Starting with 18.2.6, rollbacks are non-disruptive in nature.

When a rollback operation is performed, the Avi Controller or SEs will transition to the previous major version of the software. Selective rollback is possible for the Avi Controller and SE groups.

The following options are available: * Rollback for System * Rollback for Avi Controller only * Rollback for some or all the SE groups

Note: * Rollback of the SE Group will be to the previous version. ### Rollback for System Rollback of the system will result in the rollback of the SE(s) followed by the rollback of the Avi Controller. Use the following CLI and REST API for performing rollback for a patch version for Avi system (Controller and SE groups).

Using Avi CLI

```
[admin:controller]: > rollback system
```

Using Avi REST API

```
POST api/rollback JSON data:{?system?:true}
```

```
POST api/rollback JSON data:{?system?:true,?rollback_type?:2}
```

Rollback for Avi Controller

This interface is used to rollback the Avi Controller.

Using Avi CLI

```
[admin:controller]: > rollback controller
```

Using Avi REST API

```
POST api/rollback
```

Rollback for SE Groups

Using Avi CLI

```
[admin:controller]: > rollback segroup se_group_refs Default-Group se_patch_ref 18.2.6-9000-20191031.063017
```

```
[admin:controller]: > rollback segroup <se-group-name>
[admin:controller]: > rollback segroup seg-a
```

Using Avi REST API

```
POST api/rollback JSON data:{?se_group_uuids?: [?seg-a-uuid?]}
```

Rollback - Patch

Rollback of a patch release transitions the software to a version without the specific patch. It will NOT roll back to the previous major version.

Selective ability to rollback the patch on the Avi Controller and SE groups is available. Note: Rollback patch option is available only from Avi Vantage release 18.2.7.

This interface is used to roll back the patch and not the major version.

The followings are the available options: * System: rollback patch for Avi Controller and all SE groups * Controller: rollback patch the Avi Controller only. * SE-group: rollback patch for all or some of the SE groups.

Rollback Patch for System

Use the following CLI and REST API for performing rollback for Avi System (Avi Controller and SE groups).

Using Avi CLI

```
[admin:controller]: > rollbackpatch system
```

Using Avi REST APIs

```
POST api/rollback JSON data:{?rollback_type?:2}
```

Rollback Patch for Avi Controller

Use the following CLI and REST API for performing rollback for a patch version for an Avi Controller.

Using Avi CLI

```
[admin:controller]: > rollbackpatch controller
```

Using Avi REST APIs Add here

Rollback Patch for SE Groups

Use the following CLI and REST API for performing rollback for a patch version for an SE group.

Using Avi CLI

```
[admin:controller]: > rollbackpatch segroup <se-group-name>
[admin:controller]: > rollbackpatch segroup seg-a
```

Using Avi REST APIs

```
POST api/rollback JSON data:{?rollback_type?:2,?se_group_uuids?: [?seg-a-uuid?]}
```

Note Refer to [Additional Options for Flexible Upgrade](#) for the following additional options: * Rollback - Error Recovery * Abort Cleanup * SE Group Resume Option

Show Commands

The following show commands provide software version visibility in the system:

- `show version controller`
- `show version serviceengine`
- `show version serviceenginegroup`

The following commands provide upgrade visibility in the system.

- `show upgrade status`: Various filters will be implemented as per UI work-flow.
- `show upgrade history`: This command is deprecated.

Notes:

- The Avi Controller will be at the highest version while the SE groups may be at lower versions. Certain commands may not work due to the Avi Controller version being at the highest version.
- Due to the API version semantics, certain fields may not be available as they are deprecated in annotation.
- Due to API endpoint deprecation, some internal commands may not work.

Alerts and Events

The following events are available to provide visibility: * Image upload/delete events * Upgrade-specific events * Patch-specific events * Rollback-specific events * Rollback patch-specific events. * Failures will translate into alerts.

Additional APIs

The following GET API calls are applicable:

- The following REST API provides information about all the images present in the system.

```
Get API: api/image/
```

- The following API provides information about a specific image whose UUID is passed as a slug.

```
Get API: api/image/image-uuid
```

- Use the following API to delete the image provided if not in use.

```
Delete API: api/image/image-uuid
```

- Inventory API ?api/image-inventory This API provides the image inventory on the system. It provides filtering based on various options such as retrieve all packages for a version etc.

Additional Information

For the complete list of difference between Avi CLI commands and Avi REST APIs for Flexible Upgrades and upgrades prior to 18.2.6, refer to [Comparison Table](#).