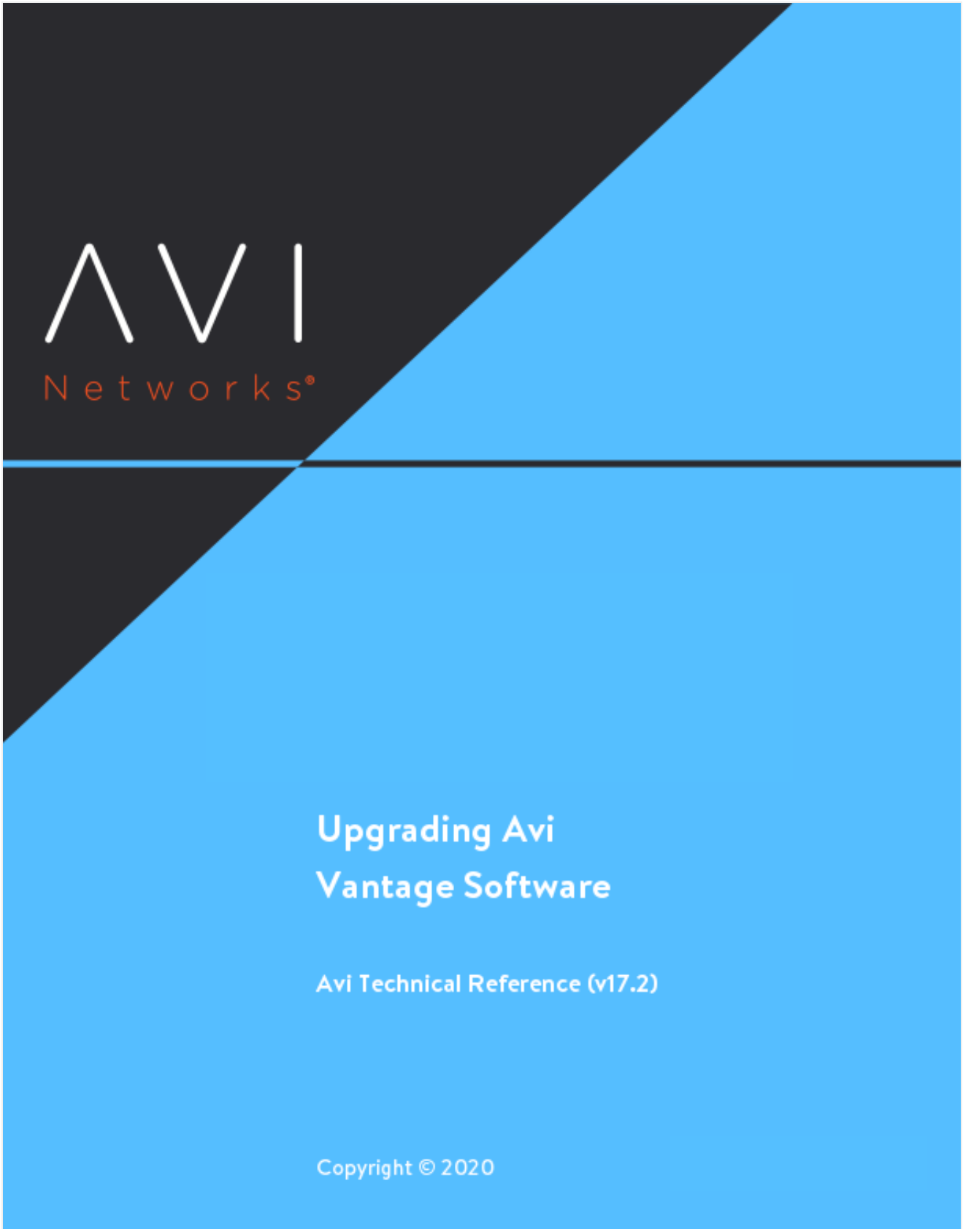

Upgrading Avi Vantage Software

Avi Technical Reference (v17.2)

Upgrading Avi Vantage Software

view online

Avi Vantage supports a simple system upgrade method wherein all Avi Controller nodes and Avi Service Engine (SE) nodes are upgraded to the newer software version in one upgrade sequence.

> *Caution: If you have installed and use any other Avi software, such as Avi Ansible Modules and Avi OpenStack drivers (Horizon, Heat, or LBaaS drivers), please coordinate to upgrade those as well when upgrading the Avi Vantage software. Doing so will ensure a successful upgrade.*

Notes: * After reading this article, if you have deployed global applications, please also read the additional valuable GSLB-specific information in the Upgrades in an Avi GSLB Environment article. For example, it is really important to upgrade the follower Controller clusters before the GSLB leader gets upgraded to the same version. If this is not done in a specific sequence it is possible that GSLB will stop working or work erratically. * See also: Patch Upgrade Process for Avi Vantage

At the onset of an upgrade, the version of the running Controller cluster is checked. If it is too distant from the release being installed, a warning is emitted and the upgrade is aborted.

If the upgrade is judged possible, the existing configuration is preserved. After the upgrade completes, Avi Vantage still has its configuration.

Management Access During Upgrade

During the upgrade process, configuration changes are blocked. The Avi Controller REST API server is switched to read-only access when the upgrade begins. Requests to get upgrade status are allowed but requests to make configuration changes are blocked.

To get status information during the upgrade, send the following request to the Avi Controller REST API server: https:///api/cluster/upgrade/status

The Avi REST API will briefly be unavailable while the Avi Controller nodes reboot.

Upgrading Avi Vantage

This section provides the steps for system upgrade using the Avi UI, CLI or REST API.

Make sure to upload the image file applicable to your Avi Controller deployment:

- Use the `controller.pkg` image file if the Avi Controller is deployed on a virtual machine (VM).
- Use the `controller_docker.tgz` image file if the Avi Controller is deployed on a bare-metal server.

> *Note: In Mesos deployments, all east-west virtual services will experience traffic disruption since they are placed only on one SE per Mesos host.*

Avi UI

- Navigate to Administration > System > System Upgrade.
- Navigate to the Avi Controller image file that applies to your deployment, and click on Upload File. The upload progress is shown.
- After the file upload is complete, click on Begin System Upgrade. The Avi Controller upgrades itself and the SEs. The progress for each phase is shown.

CLI

This section shows a representative example of the steps for upgrading.

- Download the latest version of the the appropriate image file (`controller.pkg` or `controller_docker.tgz`) from the Avi Networks portal.
- Using SCP to the Controller node's `/tmp` directory, copy the image file, which will be `controller.pkg` if deployed on a VM, `controller_docker.tgz` if deployed on bare metal. If running a 3-node cluster, copy the file only to the leader node.
- Use SSH to access the Avi Controller's CLI and enter the `shell` command:

```
Johns-MacBook-Pro:~ john$ ssh admin@10.130.150.52

Avi Cloud Controller

Avi Networks software, Copyright (C) 2013-2017 by Avi Networks, Inc.
All rights reserved.

Version:      17.1
Date:         2017-03-13 12:16:46 UTC
Build:        5124
Management:   10.130.150.52/18              UP
Gateway:      10.130.128.1                  UP

admin@10.130.150.52's password: ********


The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
Last login: Thu Apr 13 22:46:19 2017 from 10.9.0.44
admin@GSLB-East:~$ shell
Login: admin
Password: ********

[admin:GSLB-East]: > shell
[admin:GSLB-East]: >
```

- At the Avi shell prompt, enter a command such as the following. Replace `/tmp/controller.pkg` with the full path to the image file (`controller.pkg` or `controller_docker.tgz`) you downloaded.

```
> upgrade system image_path /tmp/controller.pkg
```

or

```
> upgrade system image_path /tmp/controller_docker.tgz
```

- To see the progress of the upgrade procedure:

```
show upgrade status
```

To complete the process, the Avi Controller node reboots and the SSH session is terminated.

REST API

```
POST https:///api/cluster/upgrade
DATA:

{"image_path": "controller://upgrade_pkgs/", "force": True or False, "disruptive" : True or False}
```

- For this API, the image is expected to be under `/var/lib/avi/upgrade_pkgs/` directory, with one of the two file names (`controller.pkg` or `controller_docker.tgz`). Basically, `controller://pseudo-URI` refers to the `/var/lib/avi` directory in the Controller.
  - Using SCP, to the Controller node's `/tmp` directory, copy the image file: `controller.pkg` if deployed on a VM, `controller_docker.tgz` if deployed on bare metal. If running a 3-node cluster, copy the file only to the leader node.
  - SSH to the Controller node and execute `sudo cp /tmp/ /var/lib/avi/upgrade_pkgs/`
- `force` = True or False. For every version of the image, there is a minimum compatible image. If you are upgrading from beyond that image, it will be rejected unless the `force` flag is set to True. If the `force` flag is set, it will be converted to a disruptive operation.
- `disruptive` = True or False. If you do not require the non-disruptive rolling upgrade of SEs and would rather get through upgrade quickly, you can set this flag.

How Avi Vantage Performs the Upgrade

Avi Controller Node Upgrade

After the upgrade is initiated, the Avi Controller cluster is upgraded in parallel. All the Avi Controller nodes in the cluster are updated to the newer version and rebooted in parallel. The Avi Controller waits for all the rebooted nodes to come back and re-form the cluster before continuing with the rest of the upgrade steps. If this fails, the Avi Controller will mark upgrade as aborted and roll back cleanly to the previous version.

During the upgrade process, configuration changes are blocked. The Avi Controller REST API server is switched to read-only access once the upgrade begins. Requests to get upgrade status are allowed but requests to make configuration changes are blocked.

To get status information during the upgrade, send the following request to the Avi Controller REST API server: https:///api/cluster/upgrade/status

Note: The REST API will briefly be unavailable while the Avi Controller nodes reboot and come back up.

After all the Avi Controller nodes are upgraded successfully, the upgrade is committed and rolling upgrade of the SEs begins.

Rolling Service Engine Upgrade

Rolling upgrade of the SEs is initiated by the Avi Controller, once all the Avi Controller nodes are upgraded.

The Service Engines (SEs) within each SE group are upgraded serially. The SE groups themselves are upgraded in parallel.

Virtual services are non-disruptive during SE upgrade, with one exception as listed below.

Upgrades are non-disruptive for virtual services running in:

- Elastic HA, Active-Active mode
- Elastic HA, N+M buffer mode, for virtual services scaled to two or more SEs
- Legacy Active-Standby mode

Upgrades are disruptive for virtual services running in:

- Elastic HA, N+M Buffer mode, for virtual services placed on just one SE (not scaled out)

  Note: SEs in N+M buffer mode without scaled out virtual services will be upgraded in parallel and not in series.

A note on Controller-SE communication

As a part of an SE establishing connectivity to the Controller, it sets up an SSH port-forwarding-based secure channel for the various ports on the Controller and sets up a reverse tunnel from the Controller to the SE for SSH. This is what the Controller uses to run the upgrade scripts. An internal `aviseuser` with SSH-key-based credentials is used for this. As an example, this could look as follows in the Controller for one of the SEs:

```
ssh -i /etc/ssh/id_se -p 5097 aviseuser@127.1.0.1
```

In this, the Controller's `127.1.0.1:5097` is mapped to `localhost:22` on the SE.