



**Avi Vantage Platform
CLI Guide | CLI
Guide | Knowledge
Base**

Avi Technical Reference (v17.1)

Avi Vantage Platform CLI Guide | CLI Guide | Knowledge Base [view online](#)

Avi Vantage is a robust load balancing and visibility platform for a [wide range of environments](#). Its unique architecture separates the control/management plane from the data plane. The entire load balancing fabric of distributed Service Engines (SEs aka micro-load balancers) can be managed from an Avi Controller or cluster Controller IP (in a redundant configuration).

Avi Vantage may be managed via GUI, RESTful API, or CLI. Both the GUI and CLI are built on top of the API, which means every CLI command maps to a corresponding API call(or calls) to be executed.

Interfacing to Avi Vantage via CLI Commands

Overview

In Avi Vantage there are three entities with which the authorized user may have a CLI-based conversation:

1. A Controller's Linux operating system
2. Avi Vantage processes running on a Controller
3. An SE's Linux operating system

Conversations with the first two entities are common. Conversations of the third kind are very rare, and typically undertaken by Avi Customer Support personnel for troubleshooting purposes.

CLI Conversations with the Avi Controller's Operating System

For a conversation with the Avi Controller's operating system via the `bash` Linux command line interface, either SSH to the Controller or access it via the console from an orchestrator such as vCenter. Below is a command-line example. The one argument (`user@hostname`) passed to the `ssh` command is a combination of `admin` and the IP address of the Controller, `10.144.130.195`. Every Avi Controller recognizes the `admin` user, and others can be defined if need be. The response to the password prompt is not echoed. However, in this document it is represented by `XXXXXX`.

The `bash` command-line interpreter gives access to the Controller's underlying operating system and file system. One use case would be to analyze various logs in the `/opt/avi/log` and `/var/log/upstart` directories.

```
ssh admin@10.144.130.195

Avi Cloud Controller

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

Version:      17.1.2
Date:        2017-06-05 03:14:08 UTC
Build:       9034
Management: 10.144.130.195/24      UP
Gateway:     10.144.130.1         UP

admin@10.144.130.195's password:XXXXXX
```

```
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: Tue Jul 25 18:33:37 2017 from 10.9.0.44
admin@10-144-130-195:~$
```

CLI Conversations with Avi Processes Running on the Controller

Avi Vantage-specific commands are not directly accessible from the Controller's `bash` interface. For that, enter the `Avi shell` command-line interpreter by typing the `shell` command in response to the `bash` prompt, as illustrated below. The user will be challenged for credentials.

```
shell
Login: admin
Password: XXXXX
```

Two depressions of the `TAB` key in response to the `shell`'s prompt reveals the Avi Vantage-specific commands. In the below, `TABTAB` represents the two invisible tab characters typed.

```
{% cli %}: >TABTAB attach forcedelete reprogram terminal clear gslb resync test configure import retryplacement upgrade
controller migrate rollback upload convert nsx rotatekeys upload_to_avi debug passwd scalein verifylogin delete reboot
scaleout vinfra do rediscover show watch exec redistribute switchover export reimage switcho{% endcli %}
```

The `show` command is very popular. It is used to help analyze various issues and collect information. For example, `show virtualservice my-vs` displays important data about the VS named `my-vs`.

Refer to [CLI Top-Level Commands](#) for insight into what the above-listed commands do.

To exit the Avi Vantage `shell` and return to the Controller's `bash` prompt, simply type `bash`.

```
bash
```

When jumping from Linux `bash` to the Avi Vantage `shell`, or from the Avi Vantage `shell` to Linux `bash`, return to the previous mode by typing:

```
exit
```

CLI Conversations with an Avi SE's Linux Operating System

Note: While it is possible to directly access a Service Engine's CLI, it is not recommended, and should only be used for basic troubleshooting. All configuration management should instead be done from and by the Controller.

SSH'ing to a Service Engine or executing the `attach serviceengine name-of-service-engine` command from the Avi shell places one in the SE's Linux CLI. There is no equivalent to the Controller's shell prompt; there are no show commands in the Service Engine CLI. Use the SE CLI to look into SE-specific logs in various directories, such as `/opt/avi/log`.

Navigation and Help

After dropping into the Avi shell, to see a list of available commands, press the TAB key twice.

While typing a command, TAB auto-completes the command. Double TAB returns a list of available options for the command in the left column. Most options include a brief help description, which is shown in the right column.

```
{% cli %}: > export configuration
```

```
    export configuration serviceengine
```

```
    export serviceengine ova file from controller virtualservice
```

```
    export virtual service{% endcli %}
```

Commands or parameters may require multiple words or options. If there is only a single word or option, pressing TAB auto-completes the next word in the command:

```
export configuration [TAB]
export configuration file [TAB]
WORD (required)
export configuration file mybackup
Completed writing the export configuration to mybackup
```

Other navigational commands:

- The up-arrow key cycles through and enables reuse of previously executed commands.
 - The `history` command presents commands in a list format.
 - Pipe filters results, as in
 - | `grep address`
 - | `more`
- Useful with the `watch` command.

Sub-Mode Navigation

Many Avi CLI commands contain sub-modes, which are nested sub-sections pertaining to the current command. To enter the sub-mode, enter the relevant command. Within the context of a sub-mode, changes are not committed until explicitly saved. Type `save` to exit the sub-mode while committing changes. To exit the sub-mode without saving changes, type `cancel`.

When in a sub-mode, or a nested sub-mode, the command prompt will change to reflect the current sub-mode.

```
debug virtualservice Test-VS
debug_ip
cancel
cancel
```

It is possible to enter a command which enters a sub-mode, while also adding applicable flags. This will simultaneously navigate into the sub-mode and execute the command. Subsequent commands within the sub-mode do not use the initial sub-mode command.

```
<strong>debug_ip addr 10.1.1.1</strong>
<strong>addr 10.1.1.2</strong>
<strong>save</strong>
```

The where Command

When operating within a sub-mode, multiple changes can be made to parameters. To see the current status of the configured parameters, use the `where` command.

```
<strong>debug_ip addr 10.1.1.1</strong>
<strong>addr 10.1.1.2</strong>
<strong>addr 10.1.1.3</strong>
<strong>where</strong>
Tenant: admin
+-----+-----+
| Field   | Value   |
+-----+-----+
| addr[1] | 10.1.1.1 |
| addr[2] | 10.1.1.2 |
| addr[3] | 10.1.1.3 |
+-----+-----+
```

Revealing the REST API Calls Behind CLI Commands

Any Avi CLI command may include the `--api-detail` flag, which echoes the API call (or calls) the command is performing. The command executes as it would without this flag. This can be useful when building API-driven automation scripts.

```
show serviceengine --api-detail
REST API Request
API: /api/serviceengine?owned_by_controller=True&join_subresources=runtime
```

API-echoed output may be enabled for every command executed during a single CLI session by typing the `terminal display_api_details` command, as shown below.

```
{% cli %}: > terminal display_api_details
```

```
show serviceengine REST API Request Method: GET
```

API: /api/serviceengine?owned_by_controller=True&join_subresources=runtime

```
+-----+-----+-----+-----+-----+ | Name | SE Group | Mgmt IP | Cloud |
Oper State | +-----+-----+-----+-----+ | se3 | glsbSEG |
192.168.38.56 | Default-Cloud | OPER_UP | | se1 | Default-Group | 192.168.38.52 | Default-Cloud | OPER_UP | | se2 | Default-
Group | 192.168.38.53 | Default-Cloud | OPER_UP | +-----+-----+-----+-----+
+-----+{% endcli %}
```