Avi Vantage Configuration Guide v16.1

Table of Contents

Avi Vantage CLI Guide

Avi Vantage provides a robust load balancing and visibility platform for any environment. Avi's unique architecture separates the control/management plane from the data plane. The entire load balancing fabric of distributed Service Engines (SEs), or micro-load balancers, can be managed from a Controller or cluster Controller IP (in a redundant configuration).

Avi may be managed via GUI, API, or CLI. Both the GUI and CLI are built on top of a RESTful API, which means any command within the CLI has a corresponding API that it is executing.

How to Access the CLI

To access the command line interface, either SSH to the Avi Controller or access the Controller via the console from an orchestrator such as vCenter. Some user accounts may default into Linux bash. For these accounts, enter the Avi shell by typing: `shell`

```
username@Avi-15-2:~$ shell
:>
```

Similarly, to exit the Avi shell into Linux, type: `bash`

```
: > bash
username@Avi-15-2:~$
```

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

```
username@Avi-15-2:~$ exit
:>
```

> ***Note:*** *While it is possible to directly access the CLI of a Service Engine, it is not recommended, and may only be used for basic troubleshooting. All configuration management must be done from the Controller.*

Navigation and Help

Dropping into the CLI leaves the administrator at a blank command prompt. To see a list of available commands, press the tab key twice. The commands shown are relative to the current

location within the CLI hierarchy.

While typing a command, tab will autocomplete the command. Double tab will return 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.

```
: > export
configuration       export configuration
serviceengine       export serviceengine ova file from controller
virtualservice      export virtual service
```

Commands or parameters may require multiple words or options. If there is only a single word or option, pressing tab will auto complete 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:

- Use the up arrow key to cycle through and reuse previously executed commands.
- To see the commands in a list format, use the `history` command.
- Pipe can be used to filter results, such as `| grep address` or `| more` . Useful with the `watch` command.

Sub-Mode Navigation

Many areas of the CLI 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
: debugvirtualservice> debug_ip
: debugvirtualservice:debug_ip>

: debugvirtualservice:debug_ip> cancel
: debugvirtualservice> 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.

```
: debugvirtualservice>
: debugvirtualservice> debug_ip addrs 10.1.1.1
: debugvirtualservice:debug_ip> addrs 10.1.1.2
: debugvirtualservice:debug_ip> save
: debugvirtualservice>
```

Where

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.

```
: debugvirtualservice> debug_ip addrs 10.1.1.1
: debugvirtualservice:debug_ip> addrs 10.1.1.2
: debugvirtualservice:debug_ip> addrs 10.1.1.3
: debugvirtualservice:debug_ip> where
Tenant: admin
+----------+----------+
| Field    | Value    |
+----------+----------+
| addrs[1] | 10.1.1.1 |
| addrs[2] | 10.1.1.2 |
| addrs[3] | 10.1.1.3 |
+----------+----------+
: debugvirtualservice:debug_ip>
```

API

When executing any CLI command, it may include the `--api-detail` flag. This flag will echo the API call the Avi CLI is performing, and 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 via terminal `display_api_details` .

Common Tasks

Virtual Service and Pool Creation

When creating a new Virtual Service, it will need to point to a Pool, which is expected to be pre-created. From the GUI using basic mode, this is done via a single operation. From the CLI, this is explicitly performed as two tasks.

```
: > configure pool Test
: pool> servers ip 10.1.1.100 port 80
: pool:servers> save
: pool> servers ip 10.1.1.101 port 80
: pool:servers> save
: pool> where
Tenant: admin
+------------+------------+
| Field      | Value      |
+------------+------------+
| name       | Test       |
| servers[1] |            |
| ip         | 10.1.1.100 |
| port       | 80         |
| servers[2] |            |
| ip         | 10.1.1.101 |
|port        | 80         |
+------------+------------+
: pool> save
```

Once the Pool is created, the next step is to create the new Virtual Service.

```
: > configure virtualservice Test_VS
: virtualservice> address 10.10.10.10
: virtualservice> services port 80
: virtualservice> pool_ref Test
```

Modify Pool Servers

A common task is to add and remove servers from an existing pool, or to enable or disable servers within the pool. While servers may be added to a pool without specifying a port (they will inherit the port of the pool or Virtual Service), the CLI requires the IP and Port to manipulate the server for tasks such as enable and disable.

The following commands will delete the .101 server and add a new .102 server to the pool Test.

```
: > configure pool Test
: pool> no servers ip 10.1.1.101 port 80
: pool> servers ip 10.1.1.102 port 80
: pool:servers> save
```

The following commands will enter the sub-mode for the .100 server and disable it.

```
: pool:servers> ip 10.1.1.100 port 80
: pool:servers> where
Tenant: admin
+-------+------------+
| Field | Value      |
+-------+------------+
| ip    | 10.1.1.100 |
| port  | 80         |
+-------+------------+
: pool:servers> no enabled
+---------+------------+
| Field   | Value      |
+---------+------------+
| ip      | 10.1.1.100 |
| port    | 80         |
| enabled | False      |
+---------+------------+
: pool:servers> save
```

Export / Import Configuration

The Avi Controller has a database to store its configuration information. This includes all configuration related to Tenant, Virtual Services, Pools, Policies and Accounts. Export this configuration as a JSON file for backup or to migrate parts of the config to another Avi Controller.

The configuration may be backed up or moved to another Controller cluster via the CLI. The exported configuration may be the entire system configuration, a more limited version based on the access rights of the current user and tenant of the administrator, or it may be a single Virtual Service and its child properties.

To export only a single Virtual Service and its child objects, such as Pools, use the `virtualservice` flag instead of the `configuration` flag with the export command.

The following commands will export the configuration to a file named `config_export` , SCP it to a remote location, then return to the Avi shell.

```
: > export configuration file config_export
Completed writing the export configuration to config_export
: > bash

admin@Avi-Controller:~$ pwd
/home/admin
admin@Avi-Controller:~$ ls
config_export
admin@Avi-Controller:~$ scp ./config_export root@10.1.1.1:/root
root@10.1.1.1's password:
config_export                                 100% 232KB 431.8KB/s 00:00
admin@Avi-Controller:~$ exit
:>
```

Before starting a configuration import, ensure that all Controller members of the cluster are up and the cluster leader has the following configuration:

- Admin account
- Cluster configuration
- OpenStack infrastructure (OpenStack only)

The following commands will import a backed up configuration to a Controller cluster.

```
: > import configuration file /home/admin/myconfig keep_uuid
Successfully imported the configuration file
```

You can restore this configuration information to an empty Avi Controller. The restoration steps may look like:

- Deploy 3 new Controller nodes; the image type (OVA, qcow2, ami) should match that of the original Controller node from which the configuration was exported.
- Choose one Controller as a leader and go through the initial setup page to enter the initial setup information and the redundant Controller cluster information.
- Use the CLI or API to import the configuration.

After these steps, the cluster will be back to the same running state as the original one.

Packet Capture

The Avi Controller provides a convenient mechanism to capture data plane traffic processed by Service Engines. An administrator can initiate a capture command from the Avi Controller CLI, while the actual capture occurs on the Service Engine. The packet capture is saved in the pcap format on the Controller. The Service Engine captures packets intelligently on both client side and server side of the SE. If a Virtual Service is scaled across multiple Service Engines, then all applicable SEs will participate in the packet capture. The Controller will aggregate the captures and sort the entries according to time. Once a capture is complete, it is commonly uploaded to a PC or other system for analysis with an appropriate tool such as TCPdump or Wireshark.

Enter the packet capture sub-mode for the specified Virtual Service:

```
: > debug virtualservice Test-VS
Updating an existing object. Currently, the object is:
+-------+--------------------+
| Field | Value              |
+-------+--------------------+
| uuid  | virtualservice-0-1 |
| name  | Test-VS            |
+-------+--------------------+
```

Parameters may be defined for the packet capture. By default, the capture is performed within the context of the selected Virtual Service. It is also performed on all Service Engines that are handling the VS traffic, and includes the packets from the client and server side of the SE.

`capture_params duration`	Time, in minutes. Default is unlimited.
`capture_params num_pkts`	Maximum number of packets to collect. Default is unlimited.
`capture_params pkt_size`	Packet size, or snap length, to capture. Default is unlimited.

`debug_ip addrs`	IP4 Address format
`debug_ip prefixes`	IP4 Prefix format <x.x.x.x/x>

The `debug_ip` command enters a sub-mode. This allows multiple IP addresses or IP subnets to be entered (omit the `debug_ip` for subsequent entries). Save to commit the desired IPs and return to the previous menu.

> ***Warning:*** *By default, no maximum packets or duration of time to be captured are defined. It is recommended to include a maximum packet capture as shown in the following example. Without a limit, the capture will run until filling the Service Engine disk, potentially disrupting service.*

Specify parameters, including the max number of packets to capture:

```
: debugvirtualservice> capture_params num_pkts 1000
: debugvirtualservice> debug_ip addrs 10.10.10.10
: debugvirtualservice:debug_ip> save
```

Begin capturing based on the previously configured parameters:

```
: debugvirtualservice> capture
: debugvirtualservice> save
+----------------+--------------------+
| Field          | Value              |
+----------------+--------------------+
| uuid           | virtualservice-0-1 |
| name           | Test-VS            |
| debug_ip       |                    |
|   addrs[1]     | 10.10.10.10        |
| capture        | True               |
| capture_params |                    |
|   duration     | 0 mins             |
|   num_pkts     | 1000               |
+----------------+--------------------+
```

Re-enter the packet capture sub-mode and stop an ongoing packet capture:

```
: > debug virtualservice Test-VS
: debugvirtualservice> no capture
: debugvirtualservice> save
```

Export the packet capture to a remote system that can view it via a tool such as TCPdump or Wireshark:

```
: > show debug virtualservice Test-VS capture
Please specify the destination directory: /tmp
Downloaded the attachment to /tmp/vs_virtualservice.20141205_192033.pcap
: > bash
root@Avi-CTRL:~# scp /tmp/vs_virtualservice.192033.pcap user@10.1.1.1:/tmp
```

Monitor Concurrent Connections to a Server

To track the number of a concurrent connections to a server, the server must be viewed within the context of a Pool. By prepending the `watch` command in front of the `show` , it is possible to see a near real-time updates of the connection count. Keep in mind there is some delay between the time the connection is established and the polling interval when the SE sends information back to the Controller. Cancel the `watch` with `Ctl-C` .

```
: > watch show pool Test-pool server detail | grep -e 'ip\|open_conns'
| ip_addr      | 10.1.1.17 |
|   open_conns | 50        |
| ip_addr      | 10.1.1.16 |
|   open_conns | 49        |
```

Top-Level Commands

The following are top level commands shown when pressing tab twice from the shell:

Command	Description
attach	Connect to a remote Controller or SE. Similar to SSH.
clear	Clear the statistics or value of a designated object.
configure	Create or modify a new or existing object, such as a VS, Pool, Profile…
convert	Import and convert a configuration from non-Avi load balancers.
copy	Copy a file, such as a packet capture or tech-support file.
debug	Change debug settings or perform packet captures.

Command	Description
delete	Delete an object. Some objects may have dependencies which must be resolved first.
do	Execute a show command without exiting the current location or sub-mode.
export	Back up the system configuration or a single VS configuration.
import	Import a backed up (exported) complete or VS specific configuration.
initialplacement	Specify the initial SE for a VS when using manual placement mode.
migrate	Move a VS using manual placement mode to a different SE.
purge	Remove a file, such as a packet capture or tech-support file.
rebalance	Realign the SEs handing VSs within an SE Group.
reboot	Reboot part or all of the Avi system. Can also wipe configuration.
rediscover	VMware specific: Initiate discovery of networks and VMs Disable then re-enable a VS.
restart	Disable then re-enable a VS.
scalein	Reduce by one the number of SEs handling a VS in manual placement mode.
scaleout	Increase by one the number of SEs handling a VS in manual placement mode
show	Show detailed information and stats on any Avi object.
switchto	Switch into a different Tenant.
terminal	Alter the shell’s terminal settings.
upgrade	Initiate an upgrade of the Avi system.
upload	Upload a specified tech-support debug file to Avi Networks.
verifylogin	Validate login settings to a remote orchestrator such as vCenter, APIC, or OpenStack.
watch	Updates the result of a command such as show every few

Command	Description
	seconds.

attach

Description:	Connect to a remote Controller or SE. Similar to SSH.
Example:	**attach serviceengine Avi-se-arjnz**
Top Level Flags:	
controller	Attach to a Controller shell.
serviceengine	Attach to a Service Engine shell.

configure

Description	Create or modify a new or existing object, such as a VS, Pool or Profile.
Example	`: > configure pool Test`
Top-Level Flags	
actiongroupconfig	Create or modify an Action Group Config
alertconfig	Create or modify an Alert Config
alertemailconfig	Create or modify an Alert Email Config
alertscriptconfig	Create or modify an Alert Script Config
alertsyslogconfig	Create or modify an Alert Syslog Config
analyticsprofile	Create or modify an Alert Syslog Config
application	Create or modify an Application
applicationpersistenceprofile	Create or modify an Application Persistence Profile
applicationprofile	Create or modify an Application Profile
authprofile	Create or modify an Auth Profile

cloud	Create or modify a Cloud
cluster	Create or modify a Cluster
controller	Create or modify Controller properties
healthmonitor	Create or modify a Health Monitor
httppolicyset	Create or modify an HTTP Policy Set
ipaddrgroup	Create or modify an IP Address Group
network	Create or modify a Network
networkprofile	Create or modify a Network Profile
networksecuritypolicy	Create or modify a Network Security Policy
pkiprofile	Create or modify a PKI Profile
pool	Create or modify a Pool
role	Create or modify a Role
serviceengine	Create or modify a Service Engine
serviceenginegroup	Create or modify a Service Engine Group
serviceengineproperties	Create or modify Service Engine properties
sslkeyandcertificate	Create or modify an SSL Key and Certificate Request
sslprofile	Create or modify an SSL Profile
stringgroup	Create or modify a String Group
systemconfiguration	Create or modify a System Configuration
tenant	Create or modify a Tenant
user	Create or modify a User
virtualservice	Create or modify a Virtual Service
vrfcontext	Create or modify a VRF Context
vsdatascriptset	Create or modify a VS DataScript Set

convert

Description:	Import and convert a configuration from non-Avi load balancers. Supports conversion from F5 BIG-IP Local Traffic Manager configuration. Imported Virtual Services start in a disabled state to avoid IP conflicts.
Example:	`convert bigip_configuration`
Top-Level Flags	
`bigip_ip_addr`	BIGIP IP address
`filename`	Avi config file name
`password`	BIGIP Password
`username`	BIGIP Username
`virtualservername`	Convert virtualserver. Name is of the form /partition/virtualservername

copy

Description:	Copy a file, such as a packet capture or tech-support file.
Example:	`copy file source /tmp/old destination /tmp/new`
Top-Level Flags:	
`source`	The source of the original file and path
`destination`	The destination of the new file and path

debug

Description:	Change debug settings or perform packet captures.
Example:	`: > debug virtualservice Test-VS`
Top-Level Flags:	

`controller`	Controller-specific debug options
`serviceengine`	Service Engine-specific debug options
`virtualservice`	Virtual Service specific debug options, including packet capture.

delete

Description:	Delete an object. Some objects may have dependencies that must be resolved first.
Example:	`delete pool Test-Pool`
Top-Level Flags:	
actiongroupconfig	Delete Action Group Config Delete Alert
alert	Delete Alert Config
alertconfig	Delete Alert Email Config
alertemailconfig	Delete Alert Script Config
alertscriptconfig	Delete Alert Syslog Config
alertsyslogconfig	Delete Analytics Profile
application	Delete Application
applicationpersistenceprofile	Delete Application Persistence Profile
applicationprofile	Delete Application Profile
authprofile	Delete Auth Profile
cloud	Delete Cloud
healthmonitor	Delete Health Monitor
httppolicyset	Delete HTTP Policy Set
ipaddrgroup	Delete IP Address Group
network	Delete Network

networkprofile	Delete Network Profile
networksecuritypolicy	Delete Network Security Policy
pkiprofile	Delete PKI Profile
pool	Delete Pool
role	Delete Role
serviceengine	Delete Service Engine
serviceenginegroup	Delete Service Engine Group
sslkeyandcertificate	Delete SSL Key and Certificate Request
sslprofile	Delete SSL Profile
stringgroup	Delete String Group
tenant	Delete Tenant
user	Delete User
virtualservice	Delete Virtual Service
vrfcontext	Delete VRF Context
vsdatascriptset	Delete VS DataScript Set

do

Description:	Execute a `show` command without exiting the current location or sub-model.
Example:	`do show debug flags virtualservice Test-VS`
Top-Level Flags:	
`show`	Show detailed information and stats for any Avi object.

export

Description:	Back up the system configuration or a single VS configuration.

Example:	`export configuration file /tmp/backup_config`
`export virtualservice Test-VS file /tmp/Test-VS`	
Top-Level Flags:	
`configuration`	Export the entire Avi configuration in JSON format.
`serviceengine`	Export the Service Engine OVA file from Controller for manual install.
`virtualservice`	Export a specific Virtual Service configuration file including child objects.

import

Description:	Import a backed up (exported) complete or VS specific configuration.
Example:	`import configuration file /tmp/backup_config`
Top-Level Flags:	

initialplacement

Description:	`initialplacement virtualservice Test-VS servicengine Avi-se-arjni`
Top-Level Flags:	
`virtualservice`	Specify the Virtual Service to be assigned to an SE.
`serviceengine`	Specify the name of the Service Engine to receive the VS.

migrate

Description:	Move a Virtual Service using manual placement (No Access or Read Access) mode to a different SE

Example:	`migrate`
	`virtualservice Test-VS`
	`serviceengine Avi-se-arjni`

Top-Level Flags:	
from_se_ref	Specify the name of the source SE that has the VS.
to_host_ref	An option used with `to_new_se` , specifying the host upon which to create the new Service Engine.
to_new_se	Create a new SE and migrate to it.
to_se_ref	Migrate to a specific existing SE.

purge

Description:	Remove a file, such as a packet capture or tech-support file.
Example:	`purge file source /tmp/backup_config`
Top-Level Flags:	

rebalance

Description:	In auto scale mode, sets the frequency upon which the Controller will Inspect an SE Group to see if a VS to SE mapping should be adjusted, potentially resulting in a scale in, scale out, or migration.
Example:	`rebalance interval 10 se_group_ref My_SE_Group`
Top-Level Flags:	
interval	The frequency, in minutes. Default is 5
se_group_ref	The name of the SE Group to alter.

reboot

Description:	Reboot part of all of the Avi system. Can also wipe configuration. With no flags specified, all Controllers and SEs are rebooted.
Example:	`reboot`
Top-Level Flags:	
clean	Reset the Avi system’s configuration and reboot the cluster. Consider making a backup first.
node	Reboot the virtual machine of a Controllers within a cluster.
serviceengine	Reboot a specific Service Engine. VS disruption will depend on the high availability settings for the SE Group.

rediscover

Description:	VMware specific: Initiate discovery of networks and VMs.
Example:	`rediscover vcenter My-vCenter`
Top-Level Flags:	

restart

Description:	Disable then immediately re-enable a VS.
Example:	`restart virtualservice Test-VS`
Top-Level Flags:	

scalein

Description:	Reduce by one the number of Service Engines handling a Virtual Service in manual placement mode. There must be a minimum of one SE.
Example:	`scalein virtualservice Test-VS`
Top-Level Flags:	

from_se_ref	Specify a non-primary SE to stop using for this VS.
scalein_primary	Migrate from the primary SE and discontinue use of the SE for this VS.

scaleout

Description:	Increase by one the number of SEs handling a VS in manual placement.
Example:	`scaleout virtualservice Test-VS`
Top-Level Flags:	
to_host_ref	An option used with `to_new_se` , specifying where to create the new SE.
to_new_se	Create a new SE and scale out to it.
to_se_ref	Scale out to an existing SE.

show

Description:	Show detailed information and stats on any Avi object.
Example:	`show virtualservice Test-VS summary`
Top-Level Flags:	
actiongroupconfig	Show info on an Action Group Config
alert	Show info on an Alert
alertconfig	Show info on an Alert Config
alertemailconfig	Show info on an Alert Email Config
alertscriptconfig	Show info on an Alert Script Config
alertsyslogconfig	Show info on the an Syslog Config
analyticsprofile	Show info on an Analytics Profile

apic	Show info on the APIC Graph Instances
application	Show info on an Application Folder
applicationpersistenceprofile	Show info on an Application Persistence Profile
applicationprofile	Show info on an Application Profile
authprofile	Show info on an Auth Profile
backups	Show available backup files
cloud	Show info on the Cloud
cluster	Show info on the Cluster
config-consistency-check	Show config-consistency-check status
config_events	Show info on the Event Log
configuration	Show configuration
controller	Show Controller properties
cpuusage	Show Controller CPU usage
debug	Show Virtual Service capture file
debug-log	Show Service Engine debugs
diskusage	Show Controller disk usage
events	Show info on an Event Log
file	Show files
healthmonitor	Show info on a Health Monitor
httppolicyset	Show info on an HTTP Policy Set
ipaddrgroup	Show info on an IP Address Group
jobs	Show all duration based jobs pending expiry.
License	Show info on the Controller License
Logcontrollermapping	Show mapping of log Controllers for each VS

logs-status	Show logs subsystem status
memoryusage	Show Controller memory usage
metricsmgr	Show info on a Metrics Entity Runtime
network	Show info on a Network
networkprofile	Show info on a Network Profile
networksecuritypolicy	Show info on a Network Security Policy
openstack_audit	Show OpenStack LBaaS vs Avi config audit reports
pkiprofile	Show info on a PKI Profile
placement	Show info on a Rm VRF Proto
pool	Show info on a Pool
role	Show info on a Role
serviceengine	Show info on a Service Engine
serviceenginegroup	Show info on a Service Engine Group
serviceengineproperties	Show the Service Engine properties
seupgrade	Show an ongoing SE upgrade status
sslkeyandcertificate	Show info on an SSL Key and Certificate
sslprofile	Show info on an SSL Profile
stringgroup	Show info on a String Group
systemconfiguration	Show info on the System Configuration
systemconfigurationruntime	Show info on the System Configuration Runtime
tech-support	Show full tech support
tenant	Show info on a Tenant
terminal	Show the terminal settings
transaction	Show more info on Transaction Stats
upgrade	Show upgrade status if one is in-progress

user	Show info on the User
vcenter	Show info on the specified VMs
version	Show a Controller node’s version
vinfra	Show info on the VI Datastore Contents
virtualservice	Show info on a Virtual Service
vrfcontext	Show info on a VRF Context
vsdatascriptset	Show info on the VS DataScript Set

switchto

Description:	Switch into a different Tenant.
Example:	`switchto tenant Tenant2`
Top-Level Flags:	

terminal

Description:	Alter the shell’s terminal settings.
Example:	`terminal session_timeout 240`
Top-Level Flags:	
length	Number of rows to show for pagination output. Greater thanwill pipe to more. Choose 0 for no pagination.
session_timeout	Alter the default 15 min timeout to keep an idle terminal session open.
timezone	Display timestamps in specified time zone.
unhide	Commands show additional, flags. Not recommended for casual use.

upgrade

Description:	Initiate an upgrade of the Avi system. This may be done passively (by migrating each SE while upgrading) or disruptively (fast, but it will terminate existing client connections then begin the upgrade).
Example:	`upgrade system image_path /tmp/new_file`
Top-Level Flags:	
system	Upgrade the Avi system.
ui	Upgrade the Avi UI.

upload

Description:	Generate and upload a tech-support debug file to Avi Networks.
Example:	`upload tech-support debuglogs filter exclude_logs`
Top-Level Flags	
exclude_archive	Exclude archived backups.
exclude_logs	Exclude client log files (VS logs), which may be quite large.
include_se	Include (non-significant) logs may be stored on the SEs.

verifylogin

Description:	Validate the username, password, and path to a remote orchestrator such as vCenter, APIC, or OpenStack.

Example:	`verifylogin`
	`vcenter username`
	`admin password secret`
	`vcenter_url 10.1.1.1/login`

Top-Level Flags:	
apic	Verify access to an APIC controller.

openstack	Verify access to OpenStack.
vcenter	Verify access to a VMware vCenter controller.

watch

Description:	Updates the result of a command such as show every few seconds.
Example:	`watch show pool Test-pool server detail &#124; grep -e 'ip\&#124; open_conns'`
Top-Level Flags:	
show	Select a valid `show` command syntax to update.