



Manually Validate Server Health

Avi Technical Reference (v18.1)

Copyright © 2018

Manually Validate Server Health

[view online](#)

It is often helpful to validate the response of a server while troubleshooting reasons a server may be marked down. The challenge is to ensure the test is from a specific Avi Service Engine, using the same tenant, network, and IP address. SEs have multiple network stacks, one for the control plane which uses Linux, and a second for the data plane. Simply logging into an SE and pinging a server will go out the management port and IP address, which may route through different infrastructure than the SE data plane.

For more info on health monitoring servers see:

- [Overview of Health Monitors](#)
- [Reasons a Server Can Be Marked Down](#)
- [Servers Flap Up / Down](#)
- [Health Monitor Troubleshooting](#)

Prerequisites

1. Determine the IP address of the Service Engine hosting the virtual service.
2. SSH into the Avi Controller
3. Log into the Avi shell

```
shell
```

VMware - No Tenants

1. Connect to a Service Engine's Linux shell:

```
: > attach serviceengine 10.10.25.28
```

2. Validate the current namespace:

```
admin@10-10-25-28:~$ ip netns
```

Usual output is avi_ns1, which is the default namespace.

3. Execute a static health check from this namespace. See examples at the end of this article.

VMware - Multiple Tenants

For multiple tenants on VMware, Avi Vantage does not create VRFs/namespaces by default.

1. Attach to the Service Engine Linux shell:

```
: > attach serviceengine 10.10.25.28
```

2. Execute a static health check. See examples at the end of this article.

Multiple Tenants with VRF (Provider Mode)

1. Find the namespace/VRF for the pool server:

```
: > show pool p1 detail | grep vrf_id
| vrf_id | 2
```

In this case, the vrf_id is 2, and the namespace is avi_ns2.
This information can also be obtained from:

```
: > show serviceengine 10.10.25.28 vnicdb
```

2. If there are multiple SEs, find the vrf-id on the specific SE:

```
show pool p1 detail | filter disable_aggregate se se_ref 10.10.25.28
| vrf_id | 2
```

3. Attach to the Service Engine Linux shell:

```
: > attach serviceengine 10.10.25.28
```

4. Execute a static health check from this namespace. See examples at the end of this article.

Bare Metal / Linux Cloud

For bare-metal Linux clouds, there are no namespaces, reducing the necessary steps.

1. Attach to the Service Engine Linux shell:

```
: > attach serviceengine 10.10.25.28
```

2. Execute a static health check. See examples at the end of this article.

Common Manual Server Checks

Ping:

```
root@test-se2:~# sudo ip netns exec avi_ns1 ping 10.90.15.62
PING 10.90.15.62 (10.90.15.62) 56(84) bytes of data.
64 bytes from 10.90.15.62: icmp_seq=1 ttl=64 time=26.8 ms
```

Curl:

```
root@test-se2:~# sudo ip netns exec avi_ns1 curl 10.90.15.62
curl: Failed to connect to 10.90.15.62 port 80: Connection refused

root@test-se2:~# sudo ip netns exec avi_ns1 curl 10.90.15.62:8000Welcome - Served from port 80!
```

Note: This step is not necessary when the SE is on a Docker + bare-metal setup and the Docker container itself exists in a namespace.