AVI
Networks®

**Geolocation-based
Load Balancing
Algorithm for GSLB
Members**

Avi Technical Reference (v20.1)

Geolocation-based Load Balancing Algorithm for GSLB Members

Overview

The geolocation algorithm of the Avi DNS for GSLB directs client requests to the optimal site based on the longitude and latitude of the client and GSLB sites. Each SE in the Avi DNS configuration refers to its own local copy of the geolocation database (geo-DB) to make that determination.

Operation

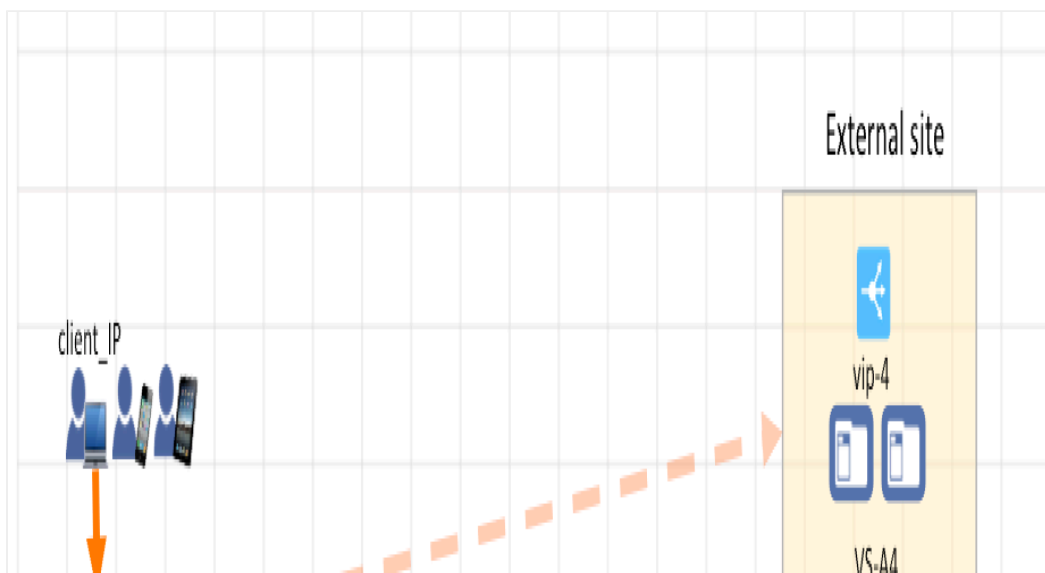Operation of the geolocation algorithm is best explained by example. Refer to figure 1.

A client with IP address = client_IP seeks service from a multi-site application implemented by four virtual services (VS-A1 through VS-A4) at four geographically separated data centers, DC1-DC4, one of which is a non-Avi load-balanced site (tersely referred to as an external site). The client?s local DNS must resolve the FQDN for the application to one of four potential addresses, vip-1, vip-2, vip-3, or vip-4. The first step is to find a DNS capable of making that translation. Either DNS-1 or DNS-2 will do. Each DNS instantiation has access to its own, identical geo-DB. Using that database, either DNS-1 or DNS-2 can translate the five IP addresses into latitude-and-longitude-based locations.

Note: The selection of DNS-2 is outside the control of Avi Vantage, and thus may have nothing to do with geographical location. We have intentionally chosen a more distant DNS to underscore this fact.
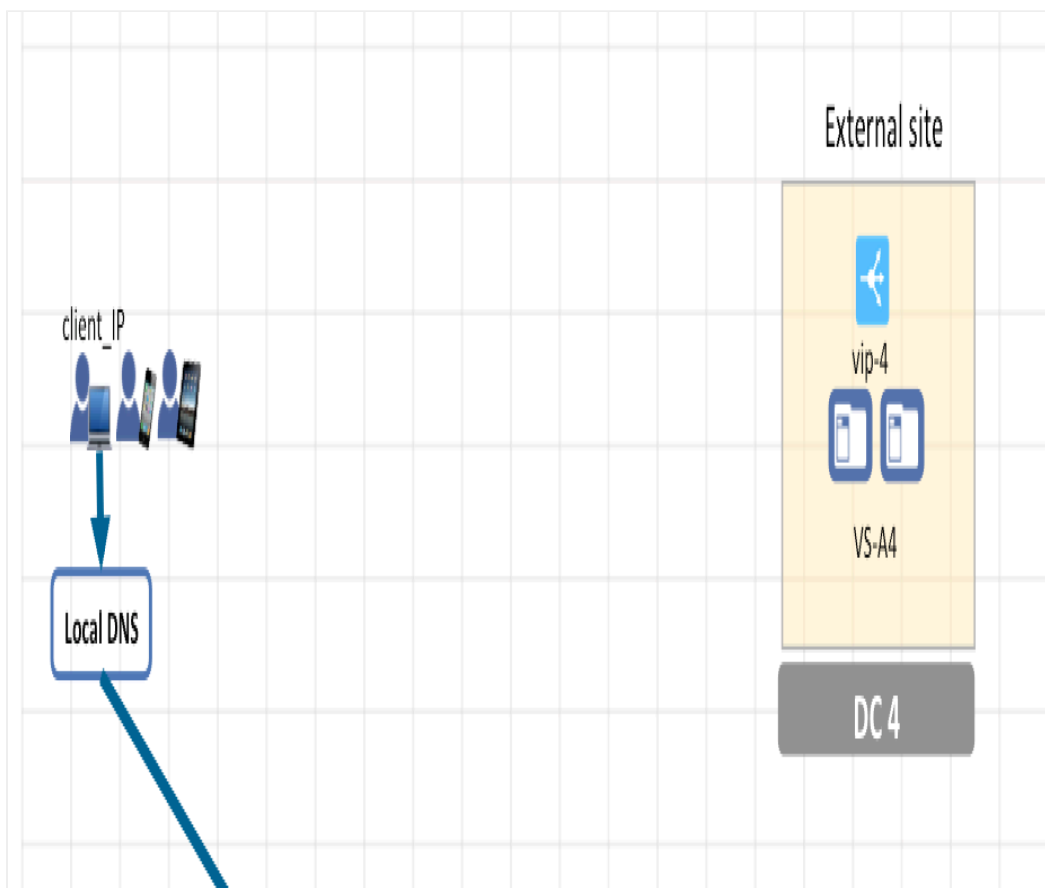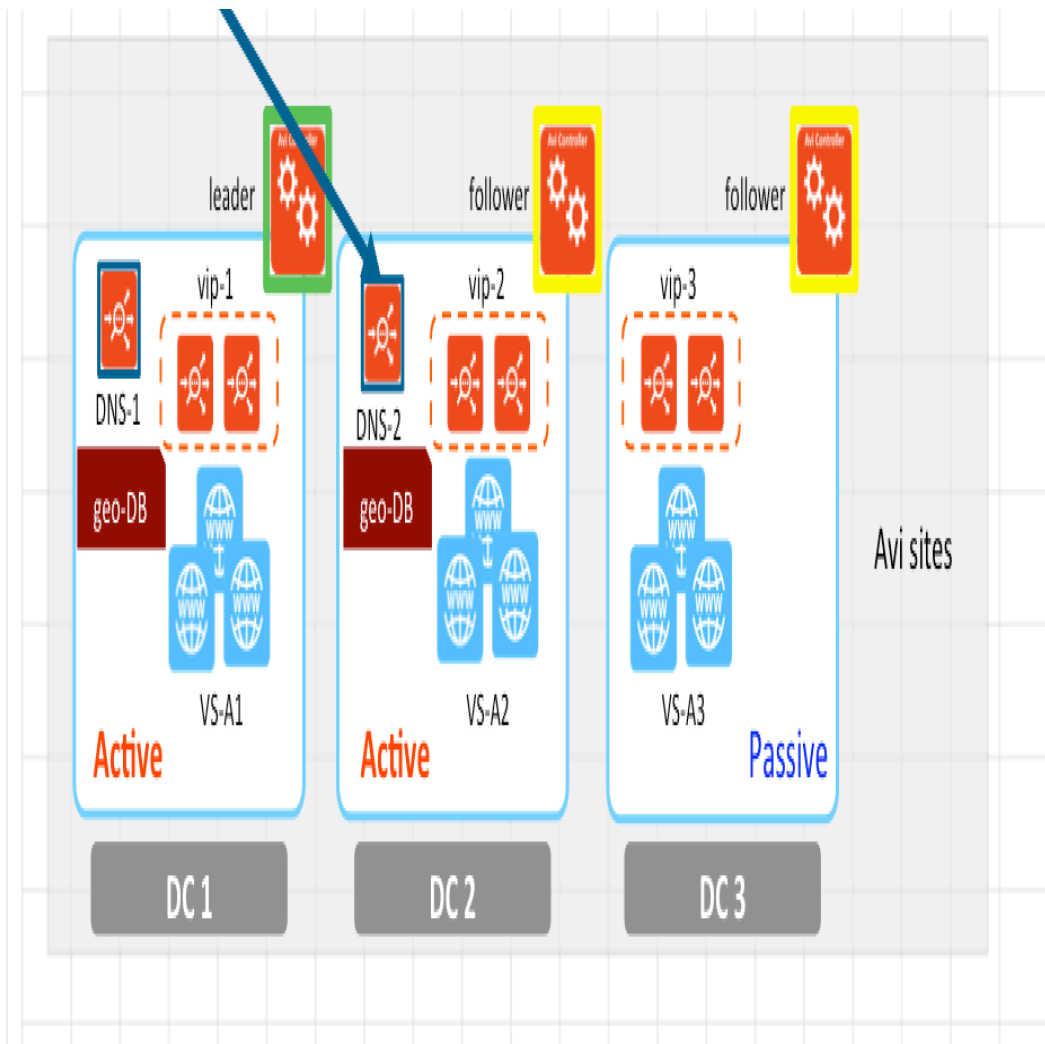
The algorithm running in DNS-2 calculates four distances, determines that dist-1 is the shortest, and therefore responds to the DNS request with vip-1. As figure 2 shows, the client thereafter directs its application request to vip-1.
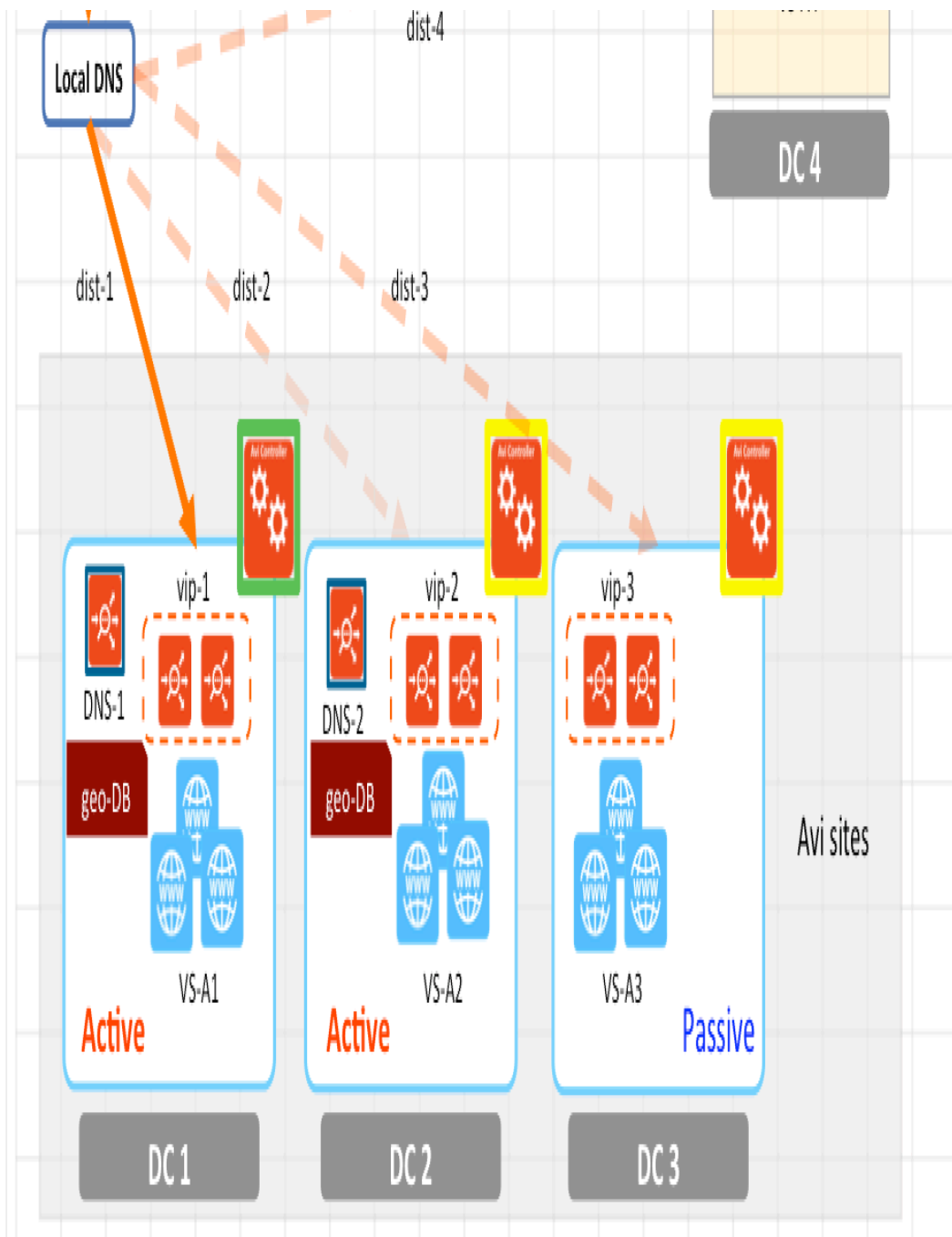
Assigning Geographic Locations to VIPs

Each GSLB service pool?s algorithm is independently set. Current choices are the round-robin, consistent hash, and geolocation algorithms. To implement the geolocation algorithm, the Avi DNS for GSLB must have a way to assign locations to GSLB sites comprising a GSLB service. The choices on a per-GSLB-service basis are as follows:

- Compute location from geo-DB (default): This method requires a look-up in a database constructed in advance. This database is described in a subsequent section of this article.

- Inherit location from site: With this methodology, each participating GSLB site is assigned a location as part of the GSLB configuration object. The GSLB service's member VS running at the site statically inherits the location associated with this site. This method can be useful when no bulk geo-DB information has been imported.

- Explicit configuration: This manual process gives the administrator full control on the most granular level without relying on a pre-populated geo-DB.

When the geolocation algorithm is chosen for a given GSLB service, member VSs having VIPs for which locations cannot yet be determined are essentially opaque. DNS responses will never include their VIPs. Instead, the geolocation algorithm will chose a VIP from among the set of VSs whose locations can be determined. If not a single member VS has a VIP whose location can be determined, the Avi DNS for GSLB will fall back to the (weighted) round-robin algorithm.

Additional Details

- Since the distance calculation is based purely on a virtual service's VIP, both Avi and external sites are candidates for VIP selection.

- DNS-2 would have returned vip-2, or vip-3, or vip-4, in geo-proximity order, had health monitoring determined such was appropriate.

- As with the round-robin algorithm, it is possible to assign weights to the VIPs. So far, we have assumed each has an equal weight, i.e., a ratio of 1:1:1:1. Were the weights changed to 1:2:1:1 (in VIP order 1, 2, 3 and 4), a DNS response other than VIP-1 could easily be imagined for this particular client. For example, assume dist-1 = 500 miles and dist-2 = 600 miles. Looking at physical distance alone, vip-1 is indeed the right choice. But assigning a weight of 2 to vip-2 causes the algorithm to compare 500 miles to 600 / 2 miles. Since 300 is less than 500 miles, vip-2 would instead be chosen. One use case for such weighting is a site whose servers are very much faster than servers at other sites. At such a speedy site, server-processing-time savings compare in magnitude to transit times over the WAN. A client's geographic location may justify routing its traffic over a somewhat longer distance for the reward of overall faster roundtrip time.

- There are times when a client's IP address does not yield an accurate client location. To address this, the administrator may choose to modify the DNS application profile to enable EDNS. If this option is selected, and an EDNS subnet extension is present in the DNS request, then it will be used instead of the client's IP address. For more information, refer to the Extension Mechanisms for DNS Client Subnet Option Insertion article.

Geolocation (Geo-DB)Database

In the lifecycle of an Avi GSLB configuration, geographic information for geolocated services is first loaded into the GSLB leader Avi Controller, in bulk or on a location-by-location basis. The data are placed into Avi geo-DB format (specified below) for download into all Avi DNS SEs in the SE group. As a reminder, this group is dedicated to DNS; no application VSs are allowed in it. The Avi geo-DB is subsequently replicated to all follower Avi DNS instantiations, so that any one of them (DNS-1 and DNS-2 in the above example) can reply to client-local DNSs with the same optimally located VIP.

Record Format

Fields in an Avi geo-DB record are sufficient to accommodate information defined in popular 3rd-party geolocation databases. A single record contains six comma-delimited fields. The first four are mandatory. A hyphen in the fifth or sixth position indicates the field is unspecified. Those fields are:

- IP address range start
- IP address range end
- Latitude

- Longitude
- Full name of city, e.g., USA/California/San Francisco
- Tag is reserved for special uses, e.g., to store a customer-defined value for region, say "West"

The ?name? and ?tag fields are used for two purposes: * They appear in logs. * They can be used to merge location entries, to make the memory usage less and lookups faster, by making the lookups coarser.

File Format and Grammar Rules

Any line starting with a leading '#' is ignored as a comment. Location record lines may not contain leading or embedded spaces. The first line must contain V.XX, where XX is the version of the Avi DB. At the time of this writing, the only supported value is V.01. The optional name field is comprised of three parts, and the parts are delimited by two slashes. Any line with a parsing error is dropped. If there are more than 95% errors, the geo-DB loading operation fails. If there are lines with duplicate or overlapping addresses, the line last specified in the file dominates, and overwrites the other entries. The following example illustrates proper syntax for IPv4 format.

```
V.01
# start_ip,end_ip,latitude,longitude,country/region/city,tag
# Hyphens denote an unspecified region and city within Australia
1.0.0.0,1.0.0.255,-33.4940,143.2104,Australia/-/-,-
1.0.1.0,1.0.1.255,26.0614,119.3061,China/Fujian/Fuzhou,-
1.0.2.0,1.0.3.255,26.0614,119.3061,China/Fujian/Fuzhou,-
# Neither the name nor the tag fields are specified for the IP range 2.0.2.0 to 2.0.3.255
2.0.2.0,2.0.3.255,26.0614,119.3061,-,-
```

Multiple files can be specified as input for source-IP-address-to-location mapping. As described later in this article, a priority mechanism determines which data should dominate if and when multiple sources stipulate a distance for a given IP address.

Conversion from other formats to the Avi format is possible. Such data is merged into one, static, composite geo-DB file in Avi format. In the first release, MaxMind format will be importable. Over time, the design may support import of data from other sources.

Clients of multi-site applications are assumed to have public or private addresses. From the perspective of Avi GSLB, they have direct network access to either public or private VIPs, but not both simultaneously. In public networks, network address translation may engage private sites, but such private IPs are transparent to the geolocation algorithm that governs public-client access. Accordingly, the geo-DB is responsible to contain one or the other kind of address, not both simultaneously.

IPv6 Support

Starting with Avi Vantage release 20.1.1, the following fields are available for an Avi geo-DB v6 record. * IPv6 address * Prefix length * Latitude * Longitude * Full name of city, e.g., USA/California/San Francisco * Tag ? is reserved for special uses, e.g., to store a customer-defined value for region, say "West"

The first four fields are mandatory. A hyphen in the fifth or sixth position indicates the field is unspecified.

IPv6 File Format

The following example illustrates proper syntax for IPv6 database format.

```
V.01
# IPv6 address, prefix length,latitude,longitude,country/region/city,tag
```

```
# Hyphens denote an unspecified region and city within United States
1::3,128,1,1,United States/-/-,testing
2::2,128,5,5,United States/-/-,testing
1::1,128,3,3,United States/-/-,testing


3::3,128,8,7,United States/-/-,testing
4::4,128,9,9,United States/-/-,testing
```

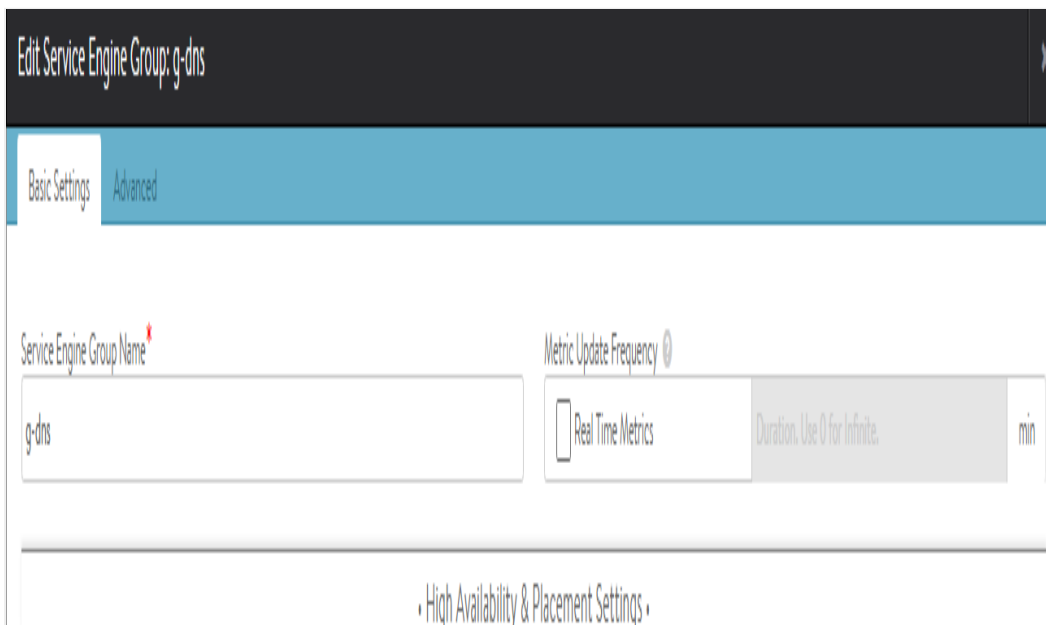# Impact on Avi Controller and Avi SE Configuration

To support the geolocation algorithm, special consideration must be given when configuring Controllers and the SEs implementing Avi DNS for GSLB.

- Memory Allocation ? An Avi geo-DB can be very large (e.g., can easily contain 3 million entries). Accordingly, the minimum recommendation for an Avi DNS SE is 8 GB. The additional knob Host Geo Profile, represented as a check box in the below screenshot, stipulates that some of that memory (2GB recommended) is to be exclusively devoted to the geo-DB. The geo profile permits a portion of SE memory to be reserved for geoDB

- Disk Allocation ? An Avi geo-DB can be very large (for instance, can easily contain 3 million entries). Accordingly, an extra 2 GB is recommended for disk size. Controller disk size likewise needs to increase beyond the norm, since it is the first place the data are assembled.

- When the Avi geo-DB is large, the DNS Service Engine needs some time to load the entries into memory.

- The CLI commands for these are discussed in the CLI section.

Configuring Geo-DB

Using Avi UI

1. Navigate to Infrastructure > GSLB > and select Upload Geo Files tab to select files that will become candidates for inclusion into the required geolocation profile. Note the format choices available in the pulldown. Starting with Avi Vantage release 20.1.1, files can contain location with IPv6 address too.

High Availability Mode ⓘ

Legacy HA
○ Active/Standby

Elastic HA
○ Active/Active   ● N + M (buffer)

VS Placement across SEs ⓘ
● Compact   ○ Distributed

Virtual Services per Service Engine ⓘ

10   Maximum   ☐ Per-app SE mode ⓘ

Extra config memory to support large Geo DB configuration.

· Service Engine Capacity and Limit Settings ·

☐ Host Geo Profile ❓

Connection Memory Percentage (slide the bar left or right) ⓘ

Connections: 50%   Buffers: 50%

2. Specify geo files to be uploaded. To create the geolocation profile, navigate to Infrastructure > GSLB > Geo Profile as shown below.
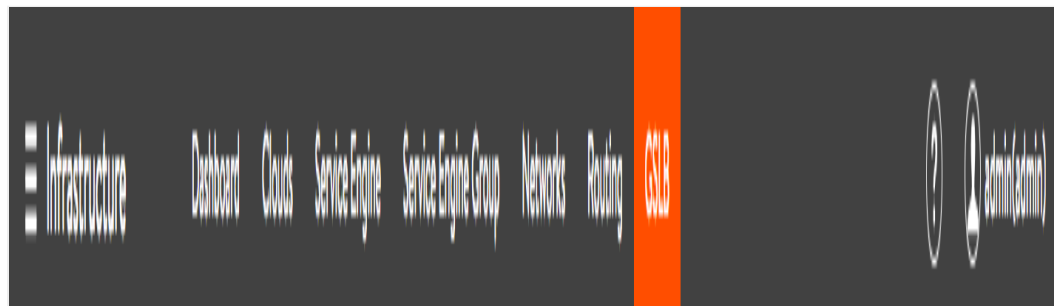
3. Click on Create to open the geo profile editor. Give the profile a name. Specify one or more previously uploaded files
    ? see below. For each files, specify a priority from 1 (lowest) to 100 (highest) and choose the file?s format from the

dropdown of supported formats. When determining the location of a given address or address range, Avi Vantage will search the files starting with the highest priority file, and pick the entry from the highest priority file.

New Geo Profile:

X

Name *

Name

Files

Filename ⊘	Priority ⊘	Format ⊘
Select File ⌄	Priority	Avi ⌄

+ Add File

Save

Using Avi CLI

1. When setting SE group properties in an environment that will use the geolocation algorithm, pay special attention to the memory configuration.

```
[admin:10-10-24-207]: > configure serviceenginegroup mySEgrp

   [admin:10-10-24-207]: serviceenginegroup> where

   Tenant: admin

   +-------+---------+

   | Field | Value   |

   +-------+---------+

   | name  | mySEgrp |

   +-------+---------+

   [admin:10-10-24-207]: serviceenginegroup> memory_per_se 8196

   [admin:10-10-24-207]: serviceenginegroup> extra_shared_config_memory 2048

   [admin:10-10-24-207]: serviceenginegroup> save

   [admin:kh-c1]: serviceenginegroup>
```
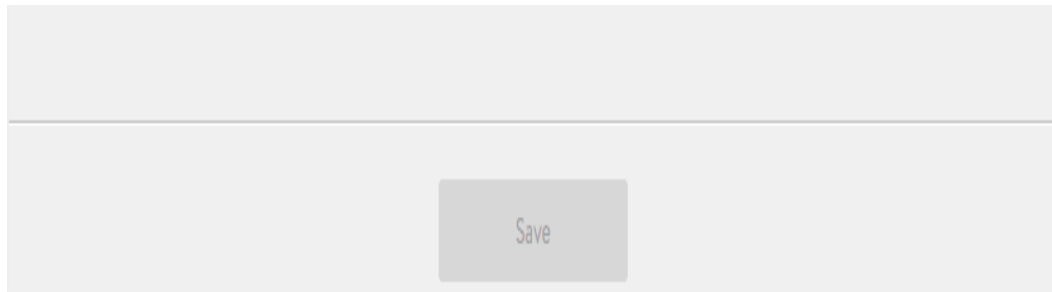
2. Upload a geoDB file(s)

```
[admin:10-10-24-207]: > upload gslbfiles [FILENAME]
```

3. Create a GSLB geolocation profile in vi editor-like fashion.

```
[admin:10-10-24-207]: > configure gslbgeodbprofile mygslbprofile

   [admin:10-10-24-207]: gslbgeodbprofile> where

   Tenant: admin

   +-------+---------------+

   | Field | Value         |

   +-------+---------------+

   | name  | mygslbprofile |

   +-------+---------------+

   [admin:10-10-24-207]: gslbgeodbprofile> [TAB][TAB]

   cancel        Exit the current submode without saving

   description   Help string not found for argument

   do            Execute a show command

   entries       (submode)

   name          A user-friendly name for the geodb profile.
```

```
    new           (Editor Mode) Create new object in editor mode
    no            Remove field
    save          Save and exit the current submode
    show_schema   show object schema
    tenant_ref    Help string not found for argument
    watch         Watch a given show command
    where         Display the in-progress object
    [admin:10-10-24-207]: gslbgeodbprofile> where
    Tenant: admin
    +-------+---------------+
    | Field | Value         |
    +-------+---------------+
    | name  | mygslbprofile |
    +-------+---------------+
    [admin:10-10-24-207: gslbgeodbprofile> new
    # description: '
<string>
  # Field Type: Optional' # entries: # - file: # checksum: '
 <string>
   # Field Type: Optional' # file_id: '
  <string>
    # Field Type: Optional' # filename: '
   <string>
     # Field Type: Optional' # format: '
    <choices: gslb_geodb_file_format_maxmind_city | gslb_geodb_file_format_avi>
      # # Field Type: Optional' # timestamp: '
     <integer>
       # Field Type: Optional' # priority: '
      <integer>
        # Field Type: Optional' # name: '
       <string>
         # Field Type: Required' # tenant_uuid: '
        <string>
          # Field Type: Required' # uuid: '
        <string>
          # Field Type: Required' # name: mygslbprofile
         </string>
        </string>
      </string>
     </integer>
    </integer>
   </choices:>
   </string>
  </string>
 </string>
</string>
```

4. **Choose the geolocation load balancing algorithm for the GSLB service pool.**

```
[admin:10-10-24-207]: > configure gslbservice gs-1
   [admin:10-10-24-207]: gslbservice> new
   # - algorithm: '
<choices: gslb_algorithm_consistent_hash | gslb_algorithm_round_robin # gslb_algorithm_geo>
  # Field Type: Optional' # consistent_hash_mask: '
 <integer>
   # Field Type: Optional' # members: # - cluster_uuid: '
  <string>
   # Field Type: Optional' # enabled: '(true | false) # Field Type: Optional' # fqdn: '
   <string>
    # Field Type: Optional' # hm_proxies: # - proxy_type: '
   <choices: gslb_health_monitor_proxy_all_members>
     # Field Type: # Optional' # site_uuid: '
    <string>
     # Field Type: Optional' # ip: # addr: '
    <string>
      # Field Type: Required' # type: '
     <choices: v4 | dns>
       # Field Type: Required' # location: # location: # latitude: '
      <float>
       # Field Type: Optional' # longitude: '
      <float>
        # Field Type: Optional' # name: '
      <string>
        # Field Type: Optional' # tag: '
      <string>
        # Field Type: Optional' # source: '
      <choices: gslb_location_src_from_geodb | gslb_location_src_user_configured # gslb_location_src_inherit_from
        # Field Type: Optional' # ratio: '
       <integer>
         # Field Type: Optional' # vs_uuid: '
        <string>
         # Field Type: Optional' # name: '
        <string>
          # Field Type: Optional' # priority: '
        <integer>
          # Field Type: Optional' # health_monitor_scope: '
         <choices: gslb_service_health_monitor_only_non_avi_members # | gslb_service_health_monitor_all_members
          # Field Type: Optional' # health_monitor_uuids:
         <string>
           # name: '
         <string>
            # Field Type: Required' # num_dns_ip: '
          <integer>
            # Field Type: Optional' # tenant_uuid: '
          <string>
            # Field Type: Required' # ttl: '
           <integer>
             # Field Type: Optional' # use_edns_client_subnet: '(true | false) # Field Type: Optional' # uui
```

```
                    <string>
                       # Field Type: Required' # wildcard_match: '(true | false) # Field Type: Optional' # groups: -
                    </string>
                  </integer>
                </string>
              </integer>
            </string>
          </string>
        </choices:>
      </integer>
    </string>
   </string>
  </integer>
 </choices:>
</string>
</string>
</float>
</float>
</choices:>
</string>
</string>
</choices:>
</string>
</string>
</integer>
</choices:>
```

5. A GSLB service using the geolocation algorithm will show the location (and any other values that might have been set using configure gslbservice).

```
[admin:10-10-24-207]: > show gslbservice gs-1
    --------------------------------------------------------------------------------+
    Field     Value
    --------------------------------------------------------------------------------+
    uuid gslbservice-90a4becd-0051-48b2-b19d-e3e2aa30f101
    name gs-1
    domain_names[1]   abcd.com
    groups[1]
    name gs11
    priority 10
    algorithm     GSLB_ALGORITHM_GEO
    members[1]
    ip    1.0.0.0
    ratio     1
    enabled   True
    controller_health_status_enabled True
    health_monitor_scope GSLB_SERVICE_HEALTH_MONITOR_ALL_MEMBERS
    enabled   True
    tenant_ref    admin
    --------------------------------------------------------------------------------+
```

```
[admin:10-10-24-207]: > show gslbservice gs-1 runtime

-----------------------------------------------------------------------+

Field     Value

-----------------------------------------------------------------------+

uuid gslbservice-90a4becd-0051-48b2-b19d-e3e2aa30f101

name gs-1

groups[1] | |

name gs11

members[1]

ip   1.0.0.0

oper_ips[1]  1.0.0.0
```

Troubleshooting

- To verify correct operation from the perspective of clients, check Avi Vantage's logs.
- If a DNS SE does not have sufficient memory to host the geo-DB, the SE event SE_GEO_DB_FAILURE is generated. If this occurs, the administrator must re-size the Service Engine by modifying the Service Engine group properties, and thereafter restart the SEs.
- For more information on troubleshooting Geolocation database issues, refer to [Troubleshooting GeoDB Issues](#).
- The filter for the following show commands now supports both v4 and v6 type addresses
- `show virtualservice g-dns-2 geolocationinfo <filter ip|start_ip>`
- `show virtualservice g-dns-2 geolocationinfo filter type [v6 | v4]`

Notes: * Based on the type chosen, the corresponding v4 or v6 output is displayed.
* 250 entries for IPv6 address and 1000 entries for IPv4 address are displayed at one time.

Other Related Articles

- GSLB Feature on Avi Vantage
- Geolocation Database