



# IPAM Provider (Google Cloud Platform)

Avi Technical Reference (v18.2)

Copyright © 2020

# IPAM Provider (Google Cloud Platform)

[view online](#)

## Overview

Avi Vantage integrates with Google Cloud Platform (GCP) for providing Internet Protocol Address Management (IPAM) services to applications running on instances in GCP. This article explains how to create a GCP profile and configure GCP IPAM.

## Prerequisites for GCP IPAM

- Set up the Avi Controller and Service Engine instances on GCP. To know more, refer to [Example Configuration of Avi Controller and SE Instances on GCP](#).
- Install Avi Vantage in a Linux Cloud. To use GCP as the IPAM provider, all the Avi Controller and Service Engine instances must run in Google Cloud. The cloud type selected within Avi should be Linux Server Cloud. To know how to set up a Linux Server Cloud, refer to [Installing Avi Vantage for a Linux Server Cloud](#).
- GCP uses a /32-based interface IP configuration. To support this, perform the following after configuring the Linux Server Cloud.

In the CLI,

```
configure serviceengineproperties
se_runtime_properties
se_handle_interface_routes

service_ip_subnets <em>subnet/mask</em>
save
save
```

Reload all the Service Engines for the configuration to take effect.

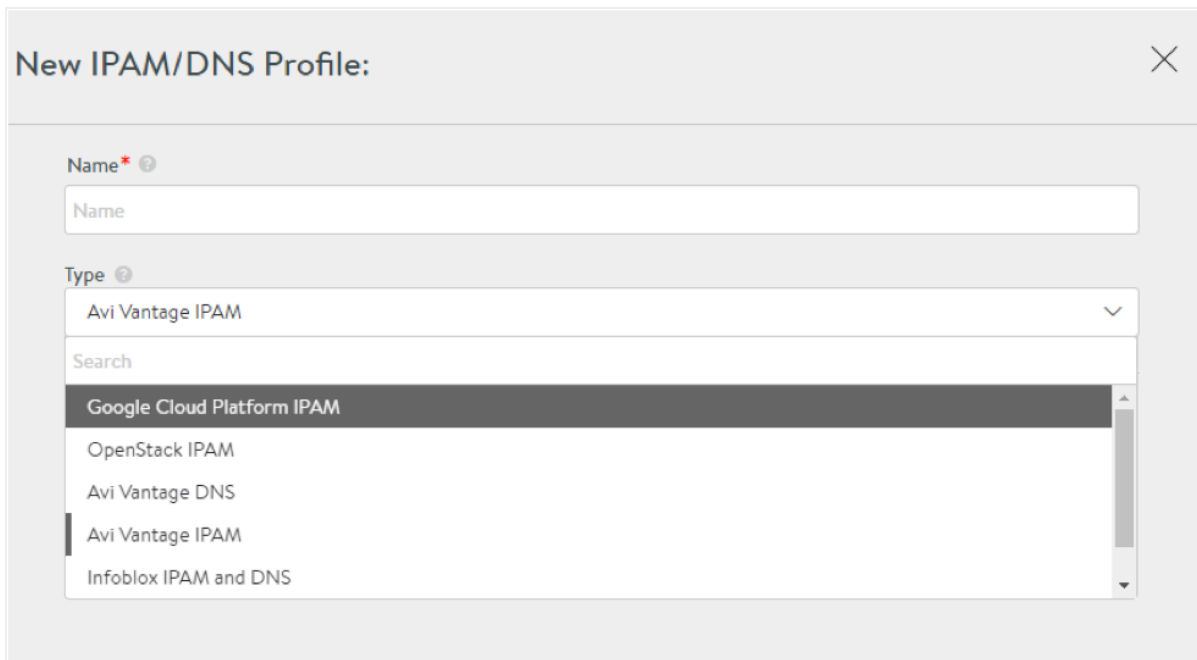
The `service_ip_subnets` configuration optimizes the Docker routes on the Service Engines. If VIP allocation is done from multiple subnets, specify all of them.

## Configuration of GCP IPAM

Create the GCP IPAM in Avi Vantage, and configure it by following the steps given below.

### Creating the IPAM Profile

1. Navigate to Templates > Profiles > IPAM/DNS profiles.
2. Click on Create.
3. Enter the IPAM Profile Name.
4. Select Google Cloud Platform IPAM as the Type as shown below.



5. Click on Save.

### Configuring GCP IPAM

1. Navigate to Infrastructure > Cloud.
2. Select Default Cloud and click on the Edit icon.
3. The IPAM that was created appears in the IPAM Profile dropdown list. Select the IPAM Profile.
4. Configure a Linux Server Cloud using the IP addresses for the Avi Service Engine instances created.

GCP IPAM has been configured, and virtual services can now be provisioned.

#### Notes

- On GCP only L3 scale out mode is supported for virtual services.
- Some of the GCP zones may require the MTU to be lowered to 1400. To set the MTU on the Service Engines,

```

configure serviceengineproperties
se_runtime_properties
global_mtu 1400
save
save
    
```

The MTU value will take effect after the Service Engines are rebooted once.

### Assigning Floating IP

Starting with Avi Vantage release 18.1.2, support for Floating IP (where a load balancer whose frontend will be a public IP allocated from GCP and the backend will be the Service Engines) is provided. Floating IP can be assigned via the CLI as shown below.



```

++Truncated Output++
+-----+
[admin:10-146-43-2]: virtualservice> vip index 1
[admin:10-146-43-2]: virtualservice:vip> auto_allocate_floating_ip
Overwriting the previously entered value for auto_allocate_floating_ip
[admin:10-146-43-2]: virtualservice:vip> save
[admin:10-146-43-2]: virtualservice> save
++ Truncated Output++
    
```

Name	Health	Address	App Domain Name	Service Ports	Pools	Throughput	# Service Engines
nlb-vs	100	55.55.55.100 (35.238.22.29)	N/A	80	xpn-nlb-cross-server-vs...	0	0

**Note:** Assigning floating IP is currently supported via the CLI only. UI support will be scoped-in in the future releases.

avi-fip-tp-73f76652475f-35-238-22-29

Frontend

Protocol	IP:Port	Network Tier
TCP	35.238.22.29:1-65535	Premium

Backend

Name: avi-...-29 Region: us-central1 Session affinity: None Health check: avi-fip-hc-73f76652475f-35-238-22-29

Instances

IP	Status
...-1	✓

### Auto-allocate Floating IP

All GCP resources will be created in the SEs project. Only auto-allocation of floating IP is supported.

```

++Truncated Output++
[2018-07-16 06:48:08,717] INFO [gcp_nlb.allocate_vip:63] old_vip_info: vip {
  addr: "55.55.55.100"
  type: V4
}
alloc_fip: false
|| new_vip_info: vip {
  addr: "55.55.55.100"
  type: V4
}
alloc_fip: true
    
```

Swagger APIs can be used to auto-allocate a Floating IP via a PUT request using the same as below.

```

"vip" : [ {
  "availability_zone" : "aeiou",
  "ip6_address" : "",
  "subnet" : "",
  "port_uuid" : "aeiou",
  "subnet_uuid" : "aeiou",
  "avi_allocated_vip" : true,
  "vip_id" : "aeiou",
  "ip_address" : "",
  "auto_allocate_floating_ip" : true, *
  "enabled" : true,
  "floating_subnet6_uuid" : "aeiou",
  "auto_allocate_ip" : true,
  "subnet6" : "",
  "floating_ip" : "",
  "floating_subnet_uuid" : "aeiou",
  "avi_allocated_fip" : true,
  "subnet6_uuid" : "aeiou",
  "floating_ip6" : "",
  "ipam_network_subnet" : "",
  "network_ref" : "aeiou",
  "discovered_networks" : [ "" ]
} ]

```

## Shared-VIP

Shared VIPs can be created on the same IP as shown in the image.

<input type="checkbox"/>	Name ^	Health ↕	Address	App Domain Name	Service Ports	Pools	Throughput	# Service Engines	
<input type="checkbox"/>	nlb-shared-vs	<span style="color: green;">100%</span>	55.55.55.100 (35.193.69.234)	N/A	8000-9000	xpn-nlb-cross-server-vs-...	0	0	
<input type="checkbox"/>	nlb-vs	<span style="color: green;">99%</span>	55.55.55.100 (35.193.69.234)	N/A	80	xpn-nlb-cross-server-vs-...	9.8 Kbps	1	

## Configure GCP IPAM for Kubernetes/OpenShift

The GCP IPAM configuration for Kubernetes involves the inclusion of fields for *Project Name* and *Network Project Name*. this is specific to the XPN case where the Network is configured in a project with "network\_project\_name" and the K8s/Avi Controllers are in a different project "project\_name" and these are in GCP region "region\_name".

Configuring GCP IPAM for Openshift/Kubernetes involves the inclusion of *Project Name* and *Network Project Name*. These fields are specific to the XPN case where the network is configured in a project with network\_project\_name and the Kubernetes/ Avi Controllers are in a different project project\_name and these are in the GCP region region\_name.

These options for GCP IPAM provider are currently supported via CLI only. The project/region details are configured via the CLI as shown below:

```
<a href="ipamdnsproviderprofile%3E%20save">admin:10-146-37-5</a>: > configure ipamdnsprov
```

```

<table>
  <colgroup>
    <col width="50%">
    <col width="50%">
  </colgroup>
  <tbody>
    <tr class="odd">
      <td align="left"><p>Field</p></td>
      <td align="left"><pre><code> Value</code></pre></td>
    </tr>
    <tr class="even">
      <td align="left"><p>uuid</p> <p>name type gcp_profile usable_network_refs[1] match_se_group_subnet use_gcp_network r</p>
      <td align="left"><p>ipamdnsproviderprofile-5*****- 4*****5 k8s-NSIPAM IPAMDNS_TYPE_GCP</p> <p>
    </tr>
  </tbody>
</table>

```