IP Address Management

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 2: Revision History

Revision Details	Release
The following sections were added: • Configuration Recommendations • IPAM Enhancements • Static IP Pool Enhancements	2021.03.0

Revision Details	Release
First introduced.	2021.01.0

Feature Description

IP Address Management (IPAM) is a method of tracking and managing IP addresses of a network. IPAM is one of the core components of the subscriber management system. Traditional IPAM functionalities are insufficient in Cloud-Native network deployments. Hence, IPAM requires additional functionalities to work with the Cloud-Native subscriber management system. The Cloud-Native IPAM system is used in various network functions, such as Session Management function (SMF), Policy Charging function (PCF), and Broadband Network Gateway (BNG).

The IPAM system includes the following functionalities to serve the Cloud Native and Control and User Plane Separation (CUPS) architecture:

- **Centralized IP Resource Management**—Based on the needs of the Internet Service Provider (ISP), the Control Plane (CP) is deployed either on a single (centralized) cluster or multiple (distributed) clusters. For multiple cluster deployments, the IPAM automatically manages the single IP address space across the multiple CPs that are deployed in the distributed environment.

- **IP Address-Range Reservation per User Plane**—For subscribers connecting to the Internet core, the User Plane (UP) provides the physical connectivity. The UP uses the summary-routes to advertise subscriber routes to the Internet core. For CPs that are managing multiple UPs, the CP reserves a converged IP subnet to the UPs. In such a scenario, the IPAM splits the available address space into smaller address-ranges and assigns it to different UPs.

- **IP Address Assignment from Pre-Reserved Address-Ranges**—When subscribers request for an IP address, the IPAM assigns addresses from the pre-reserved address range of their respective UP.

IPAM Components

This section describes the different components of the IPAM system.

IPAM Sub-Modules

The IPAM functionalities are categorized in the following sub-modules:

IPAM Server

This module manages the complete list of pools and address-space configurations. It splits the configured address-ranges into smaller address-ranges (statically or dynamically) to distribute it to the IPAM Cache modules. The IPAM server can be deployed as a centralized entity to serve a group of CN clusters or as an integrated entity within a single cluster.

IPAM Cache

This module acquires the free address-ranges from the IPAM server and allocates individual IP addresses to the IPAM clients. The IPAM cache is generally deployed in the Distributed mode running within each cluster, to communicate with the co-located or remotely located IPAM server. It is also responsible for address-range

reservation per UP and pool threshold monitoring. The IPAM server and cache modules can also run in an integrated mode.

IPAM Client

This module is tightly coupled with its respective network-function, responsible for handling request and release of individual IP address from the IPAM cache for each IP managed end-device.

Unlike the IPAM server and cache module, the IPAM client caters to use-cases specific to network-functions such as BNG, SMF, PCF, and so on.

IPAM Integration in cnBNG

The Cloud-Native Broadband Network Gateway (cnBNG) function comprises of loosely coupled microservices that provide the functionality of the BNG. The decomposition of these microservices is based on the following three-layered architecture:

1. Layer 1: Protocol and Load Balancer Services (Stateless)

2. Layer 2: Application services (Stateless)

3. Layer 3: Database Services (Stateful)

The IPAM and cnBNG integration occurs in the Application Services layer.

BNG Node Manager Application—The BNG Node Manager application is responsible for the User Plane function (UPF) management, ID and resource management, and IP address management. Therefore, the IPAM Cache is integrated as part of this microservice.

Also, the UPF uses the IPAM Client module for address-range-reservation per UPF.

BNG DHCP and PPPOE Application—The BNG-DHCP and BNG-PPOE pods are responsible for providing IP addresses to the BNG subscriber session. During session bring-up, the IP address is requested and during session bring-down, the IP address is released back. These First Sign of Life (FSOL) applications send the inter-process communications (IPC) to the Resource Manager (RMGR) component in the NodeMgr. The NodeMgr receives the IPC and invokes the IPAM component.

IPAM Server Application—Based on the deployment model, the IPAM Server runs as an independent microservice as part of the same cluster or in a remote cluster.

In standalone deployments, the IPAM Server functionality is an integral part of the IPAM Cache, that is, it runs as part of the Node Manager microservice itself.

Configuration Recommendations

This section provides the following configuration recommendations.

Pool-Size Configuration

While configuring the pool-size, the recommendation is to keep additional buffer of IPs for undisturbed subscriber churn. Consider the following recommendations:

- Released IPs become part of the quarantine queue depending on the quarantine configuration. These IPs cannot be reused until they are free.

- Due to the threshold logic, the CP automatically reserves additional chunks to the User Plane function (UPF) when requried. It based on a first-come-first-server basis.

  In a multiple UPF sharing a pool use case, whichever UPF hits the threshold first is given a chunk. Therefore, free chunks may not be available for other UPFs. That is, 100% pool utilization may not occur for a given pool in certain conditions. In this scenario, configure additional IPs accordingly or split into multiple pools per UPF.

Split-Size Configuration

While configuring the split-size of a dynamic pool, consider the following recommendations:

- Number of IPs in pool

- Total number of subscribers. That is, the number of IPs versus the percentage of the number of sessions

- Number of UPFs sharing the pool

- Number of sessions per UPF. That is, the number of sessions versus the percentage of the number of UPFs

- Quarantine time of the pool

- Churn rate. That is CPS of IP allocation, CPS of IP release

- Max routes a UPF can support (example: ASR9k supports 32 routes per loopback in Release 7.4.x/7.5.x)

Each Node Manager (nodemgr) uses per-cache as the split-size of the chunk.

Each UPF is assigned per-dp as the split-size of chunk.

Each Node Manager reserves chunks of per-cache size and further splits (if-applicable) and assigns them to a UPF.

A chunk can be in one of the following states:

- <upf-name> (means, this chunk is assigned to an UPF)

- Free:CP (means, it is free in the cache-pod, any Node Manager can use it)

- Free:NM0 (means, it is free in the Node Manager instance-0. Only this Node Manager can use it)

- Free:NM1 (means, it is free in the Node Manager instance-1. Only this Node Manager can use it)

- QT:NM0 (means, it is in quarantine in the Node Manage instance-0. This is moved to free after quarantine timer expiry)

- QT:NM1 (means, it is in quarantine the Node Manage instance-1. This is moved to free after quarantine timer expiry)

Threshold Configuration

Changing threshold configuration on-the-fly takes effect only after the next event, such as address-allocation or address-delete. On an idle-system, there may not be any immediate action.

How it Works

This section describes the call flow pertaining to the integration of the IPAM in the cnBNG.

Call Flows

This section describes the following IPAM call flows in cnBNG:

- IPAM initial sequence call flow

- IPAM call flow

- IPAM static-pool call flow

IPAM Initial Sequence Call Flow

This section describes the cnBNG initial sequence call-flow.
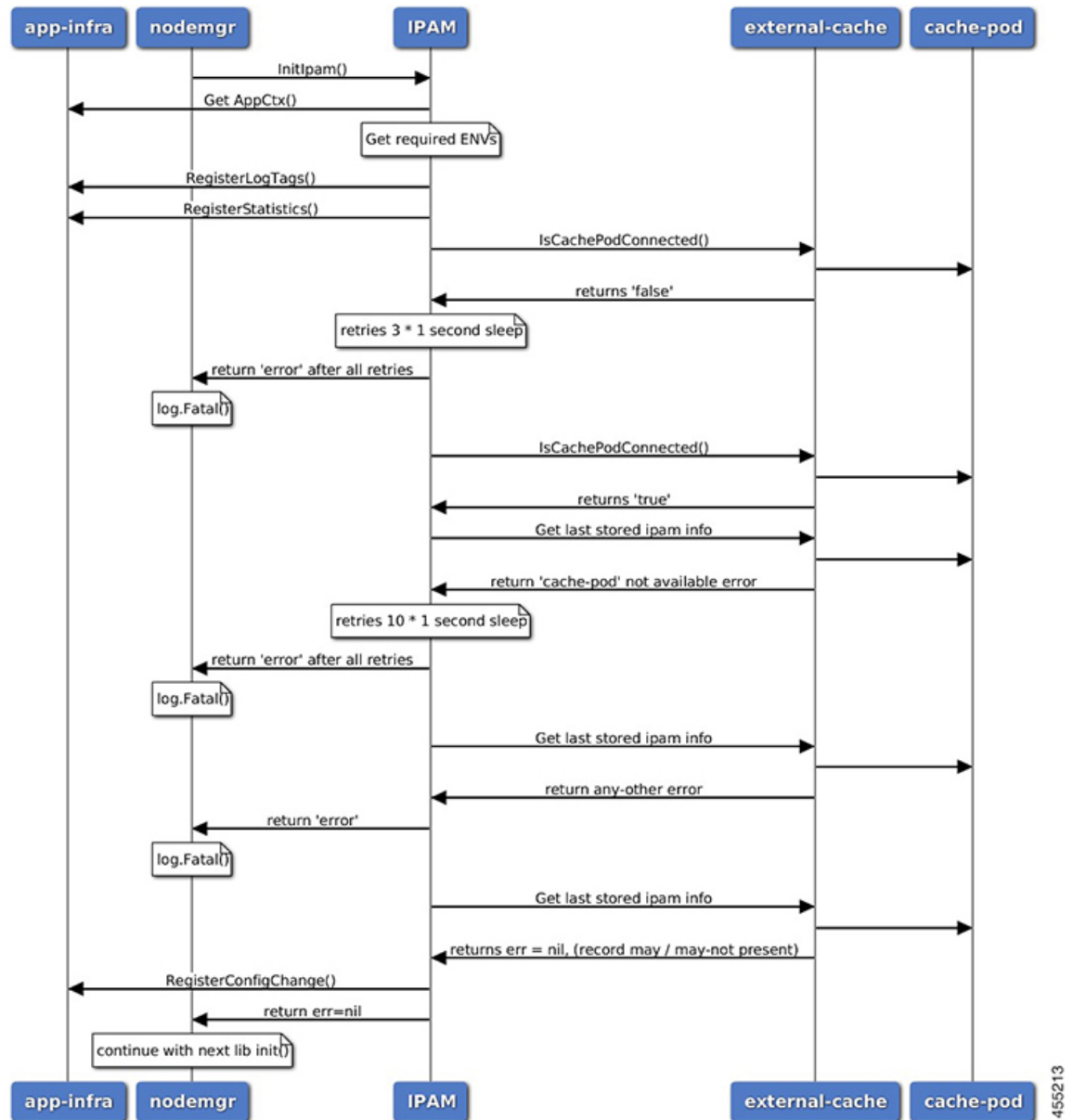
Figure 1: IPAM Initial Sequence Call Flow

Table 3: IPAM Initial Sequence Call Flow Description

Step	Description
1	IPAM reads the required environments, registers with the application infrastructure for log-tags, metrics, and database connection.
2	IPAM restores the previous state from the cache-pod, if present.
3	IPAM registers for configuration change and applies the new configuration change, if any. -change, apply new config-changes if any

IPAM Call Flow

This section describes the cnBNG IPAM call-flow.
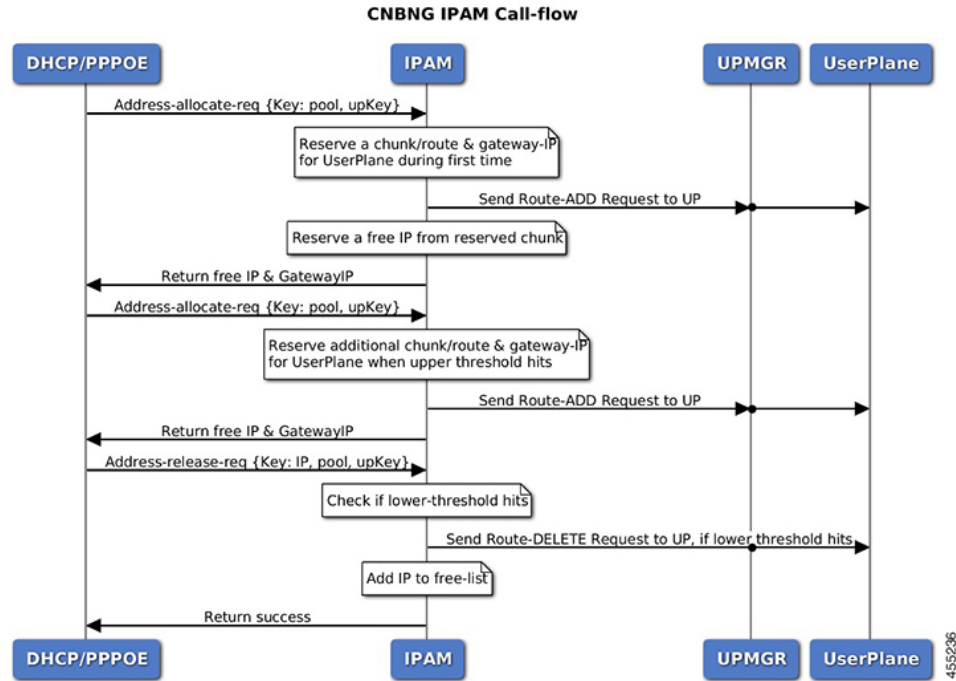
Figure 2: IPAM Call Flow

CNBNG IPAM Call-flow

DHCP/PPPOE — IPAM — UPMGR — UserPlane

Address-allocate-req {Key: pool, upKey}

Reserve a chunk/route & gateway-IP for UserPlane during first time

Send Route-ADD Request to UP

Reserve a free IP from reserved chunk

Return free IP & GatewayIP

Address-allocate-req {Key: pool, upKey}

Reserve additional chunk/route & gateway-IP for UserPlane when upper threshold hits

Send Route-ADD Request to UP

Return free IP & GatewayIP

Address-release-req {Key: IP, pool, upKey}

Check if lower-threshold hits

Send Route-DELETE Request to UP, if lower threshold hits

Add IP to free-list

Return success

DHCP/PPPOE — IPAM — UPMGR — UserPlane

455236

Table 4: IPAM Call Flow Description

Step	Description
1	IPAM receives the 'addr-alloc' request from the DHCP or PPPoE pod with pool-name, addr-type and user plane function (UPF) as input.
2	IPAM reserves a new address-range (if not already present for UPF) and sends a ROUTE-ADD message to the UPF. It waits for a success or failure response. If the receives a failure response, it removes the chunk and repeats this step.
3	IPAM reserves a free-IP from the assigned address-range and returns to the DHCP or PPPoE.
4	IPAM monitors the 'upper-threshold' for each UPF during each IP address-allocation and also has a background thread that monitors. It then assigns new address-ranges to the UPF and repeats the ROUTE-ADD flow.
5	IPAM receives the 'addr-free' request from the DHCP or PPPoE pod with pool-name, addr-type, addr or pfx, and UPF as input.
6	IPAM moves the addr or pfx first to the quarantine-list until the quarantine timer and later moves it to the free-list.

Step	Description
7	IPAM monitors the 'lower-threshold' (currently 0%) of the address-range of each UPF, removes the address-range from the UPF, and sends the ROUTE-DELETE message.

IPAM Static-Pool Call Flow

This section describes the IPAM static-pool call flow.
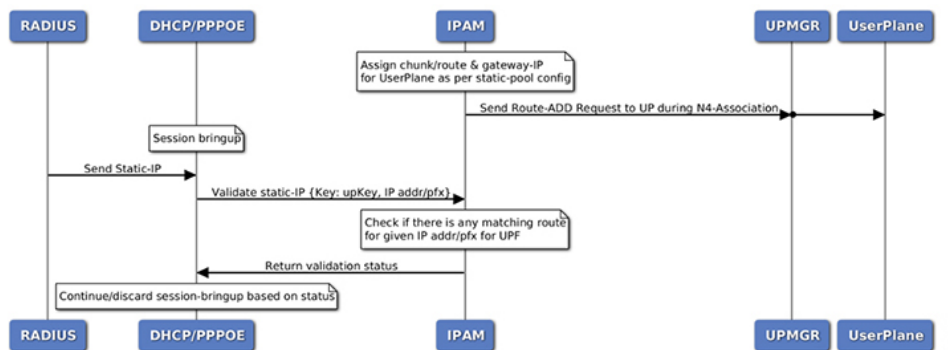
Figure 3: IPAM Static Pool Call Flow

Table 5: IPAM Call Flow Description

Step	Description
1	IPAM receives the 'addr-alloc' request from the DHCP or PPPoE pod with pool-name, addr-type and user plane function (UPF) as input.
2	IPAM reserves a new address-range (if not already present for UPF) and sends a ROUTE-ADD message to the UPF. It waits for a success or failure response. If the receives a failure response, it removes the chunk and repeats this step.
3	IPAM reserves a free-IP from the assigned address-range and returns to the DHCP or PPPoE.
4	IPAM monitors the 'upper-threshold' for each UPF during each IP address-allocation and also has a background thread that monitors. It then assigns new address-ranges to the UPF and repeats the ROUTE-ADD flow.
5	IPAM receives the 'addr-free' request from the DHCP or PPPoE pod with pool-name, addr-type, addr or pfx, and UPF as input.
6	IPAM moves the addr or pfx first to the quarantine-list until the quarantine timer and later moves it to the free-list.
7	IPAM monitors the 'lower-threshold' (currently 0%) of the address-range of each UPF, removes the address-range from the UPF, and sends the ROUTE-DELETE message.

Limitations

The IPAM feature has the following limitations:

- Duplicate IP address is not supported within a pool.

- Duplicate IP address is not supported across pools, that belong to same VRF.

- Removal of 'pool' is not supported while addresses are already assigned.

- Removal or modification of IP-address-ranges is not supported while addresses are already assigned.

- Change of 'source' field is not supported while address or prefixes are already assigned.

- Change of 'vrf-name' of pool is not supported while address or prefixes are already assigned.

- Start-address should be less than the End-address.

- Configuring addr-range split-size in wrong manner, that is, size of address-range < size-of-per-cache < size-of-dp, is not supported.

- Configuring IPv6 Address (IANA) and Prefix (IAPD) values interchangeably is not supported.

- Configuring invalid 'prefix-length' for Prefix (IAPD) range is not supported.

Configuring IPAM Feature

This section describes how to configure the IPAM feature.

Configuring the IPAM feature involves the following steps:

1. Configuring IPAM source

2. Configuring the global threshold

3. Configure IPAM address pool

4. Configuring IPv4 address ranges

5. Configuring IPv6 address ranges

6. Configuring IPv6 prefix ranges

7. Configuring the IPv4 threshold

8. Configuring the IPv6 threshold

9. Configuring IPv4 address range split

10. Configuring IPv6 address and prefix address-range split

Configuring IPAM Source

Use the following configuration to configure the IPAM source.

```
config
   ipam
```

```
      source local
        threshold { ipv4-add percentage | ipv6-address percentage | ipv6-prefix
percentage }
           commit
```

NOTES:

- **ipam**: Enters the IPAM Configuration mode.

- **source local**: Enters the local datastore as the pool source.

- **threshold { ipv4-add** *percentage* | **ipv4-address** *percentage* | **ipv6-prefix** *percentage* **}**: Specifies the threshold in percentage for the following:

   - **ipv4-add** *percentage*: Specifies the IPv4 threshold. The valid values range from 1 to 100. The default value is 80.

   - **ipv6-add** *percentage*: Specifies the IPv4 threshold. The valid values range from 1 to 100. The default value is 80.

   - **ipv6-prefix** *percentage*: Specifies the IPv6 threshold prefix. The valid values range from 1 to 100. The default value is 80.

Configuring Global Threshold

Use the following configuration to configure the global threshold.

```
config
  ipam
    threshold
       ipv4-addr percentage
       ipv6-addr percentage
       ipv6-prefix percentage
       commit
```

NOTES:

- **ipam**: Enters the IPAM Configuration mode.

- **threshold**: Enters the threshold sub-mode.

- **ipv4-add** *percentage*: Specifies the IPv4 threshold. The valid values range from 1 to 100. The default value is 80.

- **ipv6-add** *percentage*: Specifies the IPv4 threshold. The valid values range from 1 to 100. The default value is 80.

- **ipv6-prefix** *percentage*: Specifies the IPv6 threshold prefix. The valid values range from 1 to 100. The default value is 80.

Configuring IPAM Address Pool

Use the following configuration to configure the IPAM address pool.

```
config
  ipam
    address-pool pool_name [ address-quarantine-timer ] [offline ] [ static
user_plane_name ] [ vrf-name string ]
      commit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.

- **address-pool** *pool_name* **[address-quarantine-timer] [offline] [static** *user_plane_name* **] [vrf-name** *string* **]**: Configures the address pool configuration. *pool_name* must be the name of the address pool.

   This command configures the following parameters:

   - **address-quarantine-timer**: Specifies the address quarantine time in seconds. The valid values range from 4 to 3600. The default value is 4.

   - **offline**: Sets the address pool to offline mode.

   - **static** *user_plane_name*: Specifies the 'user-plane' name associated to this static-pool.

   - **vrf-name** *string*: Configures the Virtual routing and forwarding (VRF) name of the pool.

Configuring IPAM Quarantine Timer

This section describes how to configure the IPAM quarantine timer.

```
config
  ipam
    address-pool pool_name
        address-quarantine-timer quarantine_timer_value
        vrf-name vrf_name_value
        ip4
            address-range start_ipv4_address end_ipv4_address
            address-range start_ipv4_address end_ipv4_address
        !
     !
  !
```

NOTES:

- **ipam**—Enter the IPAM configuration.

- **address-pool** *pool_name*—Specifies the name of the pool to enter the pool configuration. *pool_name* must be the name of the address pool.

- **address-quarantine-timer** *quarantine_timer_value*—Specifies the value of the quarantine timer in seconds. *quarantine_timer_value* must be in the range of 4-3600 seconds. The default value is 4.

- **vrf-name** *vrf_name_value*—Specifies the name of the VPN routing and forwarding (VRF) for the pool.

- **ipv4**—Enters the IPv4 mode.

- **address-range** *start_ipv4_address end_ipv4_address*—Specifies the IP addresses for start and end IPv4 address-range.

Configuring IP Address Reservation

Use the following configuration to reserve the minimum number of addresses for each User Plane function (UPF), Node Manager, and pool.

pool.

```
config
  ipam
    instance instance_id
      min-dp-addr-size { ipv4-addr | ipv6-addr | ipv6-prefix }
      commit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.

- **instance** *instance_id*: Specifies the IPAM instance and enters the instance sub-mode. *instance_id* must be an integer. The valid value ranges from 1 to 8.

- **min-dp-addr-size { ipv4-addr | ipv6-addr | ipv6-prefix }**: Specifies the minimum number of addresses to reserve for each UPF, Node Manager, and pool.

    - **ipv4-addr**: Specifies the minimum number of IPv4 addresses to reserve.

    - **ipv6-addr**: Specifies the minimum number of IPv6 addresses to reserve.

    - **ipv6-prefix**: Specifies the minimum number of IPv6 prefixes to reserve.

Configuring IPv4 Address Ranges

Use the following configuration to configure the IPv4 address ranges.

```
config
  ipam
    address-pool pool_name
      ipv4
        address-range start_ipv4_address end_ipv4_address [ default-gateway
ipv4_address ] [ offline ]
        commit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.

- **address-pool** *pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.

- **ipv4**: Enters the IPv4 mode of the pool.

- **address-range** *start_ipv4_address end_ipv4_address* **[default-gateway** *ipv4_address* **] [offline]**: Configures the IPv4 address range with the starting and ending IPv4 address.

    - **default-gateway** *ipv4_address*: Specifies the IPv4 address of the default gateway.

    - **offline**: Sets the address pool to offline mode.

Configuring IPv6 Address Ranges

Use the following configuration to configure the IPv6 address ranges:

```
config
  ipam
    address-pool pool_name
      ipv6
        address-range start_ipv6_address end_ipv6_address [ offline ]
        commit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.

- **address-pool** *pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.

- **ipv6**: Enters the IPv6 mode of the pool.

- **address-range** *start_ipv6_address end_ipv6_address* **[offline]**: Configures the IPv6 address range with the starting and ending IPv6 address.

  [offline]: Sets the address pool to offline mode.

Configuring IPv6 Prefix Ranges

Use the following configuration to configure the IPv6 prefix ranges:

```
config
  ipam
    address-pool pool_name
      ipv6
        prefix-ranges
          prefix-range prefix_value prefix-length prefix_length
          commit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.

- **address-pool***pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.

- **ipv6**: Enters the IPv6 mode of the pool.

- **prefix-ranges**: Enters the prefix ranges mode.

- **prefix-range** *prefix_value* **prefix-length** *length*: Configures the IPv6 prefix range. *prefix_value* specifies the IPv6 prefix range.

  prefix-length *length* specifies the IPv6 prefix length.

Configuring IPv4 Threshold

Use the following configuration to configure the IPv4 threshold:

```
config
  ipam
    address-pool  pool_name
      ipv4
        threshold
          upper-threshold  percentage
          commit
```

NOTES:

- **ipam**: Enters the IPAM Configuration mode.

- **address-pool** *pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.

- **ipv4**: Enters the IPv4 mode of the pool.

- **threshold**: Enters the threshold sub-mode.

- **upper-threshold** *percentage*: Specifies the IPv4 upper threshold value in percentage. The valid values range from 1 to 100. The default value is 80.

The following is a sample configuration:

```
config
  ipam
    address-pool p1
      ipv4
        threshold
          upper-threshold 80
```

Configuring IPv6 Prefix-Range Threshold

Use the following configuration to configure the IPv6 prefix-range threshold.

```
config
  ipam
    address-pool  pool_name
      ipv6
        prefix-ranges
          threshold
            upper-threshold  percentage
            commit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.

- **address-pool** *pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.

- **ipv6**: Enters the IPv6 mode of the pool.

- **prefix-ranges**: Enters the IPv6 prefix ranges sub-mode.

- **threshold**: Enters the threshold sub-mode.

- **upper-threshold** *percentage*: Specifies the IPv6 upper-threshold value in percentage.

The following is an example configuration:

```
config
   ipam
      address-pool p3
         ipv6
            prefix-ranges
               threshold
                  upper-threshold 78
```

Configuring IPv4 Address Range Split

Use the following configuration to configure the IPv4 address range split.

```
config
   ipam
      address-pool pool_name
         ipv4
            [ no ] split-size { per-cache value | per-dp value }
            commit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.

- **-address-pool** *pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.

- **ipv4**: Enters the IPv4 mode of the pool.

- **[no] split-size { per-cache** *value* | **per-dp** *value* **}**: Specifies the size of the IPv4 range to be split for each IPAM cache allocation. The IPAM server consumes this configuration. The **no** form of this command disables the splitting of the address-ranges into smaller chunks.

  per-cache *value*: Specifies the size of the IPv4 range to be split for each Data-Plane (User-Plane) allocation. The valid values range from 2 to 262144. The default value is 1024.

  The IPAM cache consumes this configuration.

- **per-dp** *value*: Specifies the size of the IPv4 range to be split for each Data-Plane (User-Plane) allocation. The valid values range from 2 to 262144 The default value is 256.

  The IPAM cache consumes this configuration.

Configuring IPv6 Address and Prefix Address-Range-Spilt

Use the following configuration to configure the IPv6 address and prefix address range spilt.

```
config
   ipam
      address-pool pool_name
         ipv6
            address-ranges
```

```
        [ no ] spilt-size { per-cache value | per-dp value }
        commit
prefix-ranges
        [ no ] spilt-size { per-cache value | per-dp value }
        commit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.

- **address-pool** *pool_name*: Configures the address pool. *pool_name* must be the name of the address pool.

- **ipv6**: Enters the IPv6 mode of the pool.

- **[no] spilt-size { per-cache** *value* | **per-dp** *value* **}**: Specifies the size of the IPv6 range to be split for each IPAM cache allocation. The IPAM server consumes this configuration. The **no** form of this command disables the splitting of the address-ranges into smaller chunks.

  per-cache *value*: Specifies the size of the IPv6 range to be spilt for each Data-Plane (User-Plane) allocation. The valid values range from 2 to 262144. The default value is 1024.

  The IPAM cache consumes this configuration.

- **per-dp** *value*: Specifies the size of the IPv6 range to be spilt for each Data-Plane (User-Plane) allocation. The valid values range from 2 to 262144 The default value is 256.

  The IPAM cache consumes this configuration.

IPAM Enhancements

This section lists the following IPAM enhancements.

IPAM Quarantine Timer

The IP quarantine logic enhancements are as follows:

- The maximum quarantine configuration is increased to 1 hour (Range: 4 to 3600 seconds).

- If the configured quarantine time is <= 15min, additional buffer of 60 seconds is added to the configured quarantine time.

- If the configured quarantine time is > 15min, additional buffer of 5 minutes is added to the configured quarantine time.

- Default quarantine time processing thread interval is changed from 5 to 60 seconds.

- The IP is moved to the free-list after ~(configured-qTime + buffer + delay-from-qt-thread-processing).

- Upon Node Manager pod restart, quarantine time of all older IPs in the quarantine time-queue is reset and will restart from beginning.

- After Node Manager pod restart, all IPs released as part of reconciliation are moved to the quarantine-queue before moving to the free-bitmap (this includes pre-reserved IPs).

Address-Range Level Quarantine

If an address-range is removed from the UPF after releasing all the IPs in a proper manner (that is, each released IP went through quarantine time) then the address-range is moved directly to free-list.

If an address-range is removed from the UPF due to the UPF-release with some of the addresses allocated, then the complete address-range is put under quarantine for the configured time and then moved to free-list.

The **show ipam pool** command displays quarantine-chunks with a special 'alloc-context'.

Pool and UPF Threshold Monitoring

The UPF threshold monitoring enhancements are as follows:

- **Upper threshold:** Default = 80%, configurable. This is used to add new chunks to the pool or UPF.

- **SafeCutOff:** Default = (upper-threshold-5%), not-configurable. After hitting upper-threshold, new chunks are allocated to the pool or UPF to bring down the current-utilization to safecutoff level , that is, upper-threshold – 5%.

- **Lower threshold:** Default = 90% of upper-threshold, not-configurable. This is used to remove a chunk from the pool or UPF.

Each Node Manager runs a pool level threshold monitoring. When a chunk is assigned to the UPF, the Node Manager checks the pool-threshold hit and reserves additional chunks from the cache-pod for future use.

For pool threshold calculation, the total number of IPs left in free-chunks are considered; not the actual number of allocated IPs on an assigned chunk. That is, after a chunk is assigned to the UPF, it is considered as fully used for pool-threshold monitoring purpose. A complete free address-range can be released back to the cache-pod based on lower-threshold calculation.

For UPF threshold monitoring, the actual number of total IPs and allocated IPs are considered; more chunks are reserved for the UPF when the upper-threshold hits. The Node Manager adds the route to the UPF whenever a new chunk is assigned to it due to the threshold hit. For performance reasons, the route is not deleted if it was added in at the last minute.

The upper threshold is configurable (default=80%), when this threshold hits, new chunks are added until the current-utilization falls back to the safe-cutoff level. That is, 75% is safe cutoff if the upper-threshold is 80%.

Lower threshold is 90% of the upper-threshold. Thati is, if the upper-threshold is 80%, then the lower-threshold is 72%, a chunk can be removed from the UPF only when the remaining threshold is below 72%. Otherwise, the chunk remains in the UPF assigned list. This logic is applied to avoid frequent route-add and route-delete operations around boundary condition. The UPF threshold monitoring is triggered during events such as address-allocate, address-release, and config-change. On idle-system, the behavior may differ, however, in a running system, the threshold calculation occurs regularly.

Marking a pool or address-range as offline overrides the lower-threshold logic. That is, if an offline chunk is completely free, it is removed from the UPF irrespective of the lower-threshold calculation.

Multiple Replica Handling

IPAM is part of the Node Manager (nodemgr) pod. A maximum of two nodemgr pods are supported per BNG cluster.

During UPF-registration, one of the nodemgr pod gets all the static-pool-routes for the UPF and all the dynamic-pool-routes from both the nodemgr pod if anything is allocated earlier and programs it.

During IP-allocation, the IPC request goes to one of the nodemgr pods. If no routes were assigned earlier, a new route is assigned and if successful, an IP is returned to FSOL. Even if one nodemgr pod goes down, the

other nodemgr can handle IP-allocations, provided enough IPs are available. Chunks that are reserved by one nodemgr cannot be used by the other nodemgr for address-allocations.

During IP-release, the IPC request should go to the IP owner nodemgr as best-effort. If the IPC fails due, then the IP become stale on the IPAM. During nodemgr bring-up, the CDL reconciliation occcurs, which recovers the stale IPs. In addition, a new CLI is added **reconcile-ipam** to manually trigger IPAM-CDL reconciliation on a need basis. This command should be executed only during maintenance because it is a heavy operation.

During the UPF release, the N4 release comes to one of the nodemgrs. It sends an internal-IPC to the other nodemgr and both clean-up all the routes assigned to the respective UPF. If one of the nodemgr is down during that time, the other nodemgr takes over the content and releases the chunks on behalf of its peer.

Static IP Support

Feature Summary and Revision History

Summary Data

Table 6: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	*Cloud Native BNG Control Plane Command Reference Guide*

Revision History

Table 7: Revision History

Revision Details	Release
First introduced.	2021.03.0

Feature Description

IPAM is the core component of the subscriber management system. Traditional IPAM functionalities prove insufficient in the Cloud Native network deployments. Therefore, IPAM requires additional functionalities to work with the Cloud Native subscriber management system.

The Static Pool Enhancements provide more functionalities on the cnBNG using IPAM. These functionalities are as follows:

- Supports addition of a new static-pool or static-addr-range dynamically when system is running.
  - Send routes when User Plane function (UPF) is already associated.

• Supports removal of a new static-pool or static-addr-range dynamically when system is running

  • Mark the static-pool or static-addr-range offline.

  • Clear subscribers, manually.

  • Delete the configuration. IPAM sends the route-delete command to the UPF if it is already associated.

✎

Note   Changing the UPF name directly in the static IP pool is not supported. To change the UPF name, gracefully delete the static IP pool and readd the new UPF.

Note the following guidelines while configuring a static IP pool:

• An address pool is marked as 'static' during configuration. A given address pool can be either configured as 'dynamic' or 'static' mode only. It cannot be modified while the system is running.

• Each static pool is associated to an Use Plane function (UPF). The IPAM configures all the address-ranges as routes on the UPF. The recommendation is to use the 'no-split' configuration to avoid having too many splits and routes.

• The IPV4 address ranges must be configured with the 'default-gateway' value, because unlike dynamic address-range, IPAM cannot assign 'default-gateway' on its own. Also, the static address-ranges must not be split further. Therefore, use the 'no-split' configuration.

Configuring Static IP Support

This section describes how to configure the Static IP support feature.

Configuring the Static IP Support involves the following step.

Configuring Static IPv4 Address Ranges

Use the following configuration to configure the Static IPv4 address ranges.

```
config
  ipam
    instance instance_id
      source { local | external }
      address-pool pool_name
      vrf-name vrf_name
      static enable user_plane_name
      ipv4
        split-size no-split
        exit
      ipv4
      address-range start_ipv4_address end_ipv4_address
        default-gateway ipv4_address
        exit
```

```
        exit
    exit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.

- **address-pool** *pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.

- **vrf-name** *vrf_name*: Specifies the name of the VPN routing and forwarding (VRF) for the pool.

- **static enable** *user_plane_name*: Configures static IP details. Sets the specified User Plane (UP) as static. *user_plane_name* is the specified UP for this static pool.

- **ipv4**: Enters the IPv4 mode of the pool.

- **split-size no-split**: Specifies that the address-ranges should not be into smaller chunks.

- **address-range** *start_ipv4_address end_ipv4_address*: Configures the IPv4 address range with the starting and ending IPv4 address.

- **default-gateway** *ipv4_address*: Specifies the IPv4 address of the default gateway.

- IPAM does only route-validation for Static IP. Validate if UPF has a route for a given static IP address or prefix. Individual address-management is not performed.