High Availability Services

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Product(s) or Functional Area	AMF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled - Always-on
Related Documentation	Not Applicable

Revision History

Table 2: Revision History

Revision Details	Release
Sub-feature introduced. SCTP High Availability Service	2022.01.0
First introduced.	2021.04.0

Feature Description

High Availability (HA) is the ability of a system to operate continuously for a designated time without significant down time.

HA uses two pods, one as active and other one as standby. Whenever the active pod goes down, the standby pod becomes active and handles the traffic.

This feature supports the following HA services:

- AMF
- NGAP and NAS
- SCTP

AMF High Availability Service

Feature Description

The High Availability feature ensures the following functionalities for AMF-service:

- No session loss when AMF-service pods get killed or restarted.
- During restart, the AMF-service pods don't:
  - Fail any procedures
  - Increase in call processing time
  - Result in call failure of the retried calls
  - Restart or crash other pods
  - Downgrade the performance

NGAP and NAS High Availability Service

Feature Description

The AMF protocol pod maintains the security context cache, NAS UL, and DL counters. Whenever this information is modified in the cache, the same information gets replicated to the peer protocol pod to ensure high availability.

The AMF protocol pods determine among themselves who is the leader by using the Etcd for electing a leader. The leader information gets registered in the topology management module in the Etcd. The leader selection upgradation helps with replicating the security context cache to the other AMF protocol pod. If the leader pod goes down, the other (follower) pod becomes active and handles the traffic. The follower pod works with the replicated security context cache, UL, and DL counters from the leader.

The AMF-SCTP and the AMF-service pods query the leader information for the AMF protocol pod before making any IPC call. When the leader pod goes down, the other pod gets selected as a leader and the subsequent IPC requests goes to the selected protocol pod.

If a pod comes up, the security context cache gets synced with the peer before the pod becomes ready.

This feature supports two AMF protocol replicas.

Note Supported maximum two replicas (recommended) for high availability.

If both protocol pod replicas go down back to back or together, the security context data gets lost.

Feature Configuration

To configure this feature, use the following configuration:

```
config
   instance instance-id instance_id
      endpoint ngap replicas replica_count
      end
```

NOTES:

- **endpoint ngap replicas** *replica_count*—Specify the number of NGAP replicas per node.

Configuration Example

The following is an example configuration.

```
config
   instance instance-id 1
      endpoint ngap replicas 2
      end
```

SCTP High Availability Service

Feature Description

SCTP uses virtual IP (VIP) to support HA. This feature supports two SCTP endpoints.

The SCTP pod starts and listens on VIP. If one SCTP pod goes down, traffic moves to the other SCTP pod using VIP.

Feature Configuration

To configure this feature, use the following configuration procedure:

1. Configure the k8 node labels, on which the SCTP pod should run.

   k8 label sctp-layer key smi.cisco.com/node-type value sctp

Note The label must have a minimum number of two K8 nodes for active or standby pods to work.

2. Configure the two replicas as active and standby pod for SCTP. The active pod receives the traffic.

```
config
   instance instance-id instance_id
      endpoint sctp
         replicas replica_count
         end
```

NOTES:

- **replicas** *replica_count*—Specify the number of SCTP replicas per node.

3. Configure the VIP for IPv4 and IPv6 using the following commands:

```
config
   instance instance-id instance_id
      endpoint sctp
        vip-ip ipv4_addressoffline { vip-interface interface_name | vip-port
 port_number }
          vip-ipv6 ipv6_address { offline  | vip-ipv6-port ipv6_port_number
}
          end
```

NOTES:

- **vip-ip** *ipv4_address* **[offline | vip-interface** *interface_name* **| vip-port** *port_number* **]**—Specify the IPv4 address of the pod on which VIP is enabled, interface, and the port number. This configuration marks VIP-IP as offline (standby).
- **vip-ipv6** *ipv6_address* **[offline | vip-ipv6-port** *ipv6_port_number* **]**—Specify the IPv6 address of the pod on which VIP is enabled. This configuration marks VIP-IP as offline (standby) if you specify as offline.

Configuration Example

The following is an example configuration.

```
config
   instance instance-id 1
      endpoint sctp
         replicas 2
         instancetype IPv6
         vip-ipv6 0001:000:00c1::4 vip-ipv6-port 1000
         end
```