



# Multiple AMF Instances Support

- [Feature Summary and Revision History, on page 1](#)
- [Feature Description, on page 1](#)
- [Feature Configuration, on page 2](#)
- [Troubleshooting Information, on page 3](#)

## 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
First introduced.	2021.04.0

## Feature Description

Multiple AMF instances enable the administrators to scale up or scale down the instances to meet the evolving capacity requirements. The AMF architecture supports seamless inclusion and exclusion of the AMF instances in the AMF framework.

## Considerations

This feature has the following considerations in this release:

- If the AMF deployment scenario has multiple servers, the servers must be labeled. The labeling is helpful when one of the server nodes is faulty or out of service.

You can label the servers as:

- Server1: Labeled for protocol and service layer pods
  - Server2: Labeled for service and datastore layer pods
  - Server3: Labeled for datastore and protocol layer pods
- When multiple instances of AMF are deployed on the servers, and these servers do not have sufficient hardware resources (CPU cores, memory), then you can reset the default pod limit prescribed by Kubernetes.

You can scale the number of pods by using the following configuration in the SMI Deployer CLI:

```
clusters cluster name node-defaults k8s max-pods number_of_pods
```

## Feature Configuration

Configuring this feature involves the following steps:

1. Configure the instance ID for the AMF instance. For more information, refer to [Configuring the AMF Instance ID, on page 2](#).
2. Associate the AMF instances to a common CDL. For more information, refer to [Associating the AMF Instances to CDL, on page 3](#).
3. Configure the notifications that CDL sends to the AMF instances. The notifications are invoked when events such as timer expiry occurs. For more information, refer to [Configuring the CDL Notifications, on page 3](#).

## Configuring the AMF Instance ID

To configure the AMF instance ID, use the following configuration:

```
config
  deployment
    logical-nf-instance-id instance_id
  end
```

NOTES:

- **deployment**—Configure the deployment parameters.
- **logical-nf-instance-id** *instance\_id*—Specify the unique instance ID for the AMF instance.
- Ensure to configure the instance ID for each AMF instance.

## Associating the AMF Instances to CDL

To associate the AMF instances, use the following configuration:

```
config
  datastore
    session-db
    endpoints
      datastore-ep-session.cdl_namespace.svc.cluster.local
      port port_number
    end
```

NOTES:

- **datastore**—Configure the datastore parameters.
- **session-db**—Configure the session database parameters.
- **endpoints**—Configure the endpoint parameters.
- **datastore-ep-session.cdl\_namespace.svc.cluster.local** —Specify the CDL namespace.
- **port port\_number**—Specify the port number of the CDL pod.

## Configuring the CDL Notifications

To configure the CDL notifications, use the following configuration:

```
config
  datastore
    notification-ep { host host_address | port port_number }
  end
```

NOTES:

- **datastore**—Configure the datastore parameters.
- **notification-ep { host host\_address | port port\_number }**—Specify the VIP IP address and port number of the AMF instance to which the CDL must send the notification. Ensure that the VIP IP and port number are unique for each AMF instance.

## Troubleshooting Information

This section describes troubleshooting information for this feature.

### Problem

In the multiple AMF deployment scenario, the secondary pods cannot be brought up when the master node has utilized the default pod limit. The secondary pods failed to be up due to one of the following reasons:

- Pods are in the pending state.
- The node didn't match the pod and node affinity or the antiaffinity rules.

### Resolution

In circumstances when all the pods are utilized, you can increase the number of pods in the Kubernetes cluster. To configure the Kubernetes maximum pod count, use the following steps:

### On an existing AMF deployment

1. In the SMI Deployer CLI, use the following configuration:

```
clusters node-defaults k8s max-pods maximum_pods
```

#### NOTES:

- a. **max-pods** *maximum\_pods*—Specify the maximum number of pods per node. Default is 256. Must be an integer in the range of 10-2000.
2. Assign labels to the nodes within the cluster and sync the changes using the following command:

```
clusters nodes actions sync run
```

3. Delete the Istio directory from the master node using the following command:

```
rm -rf /var/lib/smi/istio/
```

4. Synchronize the clusters to reflect the configuration using the following commands:

```
clusters cluster_name actions sync run reset-k8s-nodes true debug true
```

```
clusters cluster_name actions sync run sync-phase opscenter debug true
```




---

**Important** The synchronize procedure erases the AMF's day-1 or N configuration.

---

### On a new AMF deployment

1. In the SMI Deployer CLI, use the following configuration:

```
clusters node-defaults k8s max-pods maximum_pods
```

#### NOTES:

- a. **max-pods** *maximum\_pods*—Specify the maximum number of pods per node. Default is 256. Must be an integer in the range of 10-2000.
2. Synchronize the clusters using the following commands:

```
clusters cluster_name actions sync run force-vm-redeploy true debug true
```

For information on the SMI configuration, see *Ultra Cloud Core Subscriber Microservices Infrastructure - Operations Guide*.