← Resumen

Referencia de Configuración de OmniNSSF

La configuración del operador se lee desde la clave de entorno de la aplicación `:omnissf`. La configuración en tiempo de ejecución se proporciona a través de `config/runtime.exs`, que mapea las variables de entorno del sistema operativo a las claves de configuración al inicio. La topología de slice (lista NSI, NSSAI permitidos, mapeo de conjuntos AMF, NSSAI configurados) se proporciona a través de la configuración estructural de la aplicación (ver Configuración de Topología de Slice) y se ajusta en tiempo de ejecución a través de la API OAM.

Configuración en Tiempo de Ejecución (`config/runtime.exs`)

```
config :omnissf,
  sbi_scheme: System.get_env("SBI_SCHEME", "http"),
  sbi_addr: System.get_env("SBI_ADDR", "127.0.0.14"),
  sbi_port: String.to_integer(System.get_env("SBI_PORT", "7777")),
  nrf_uri: System.get_env("NRF_URI", "http://127.0.0.1:7777"),
  mcc: System.get_env("MCC", "999"),
  mnc: System.get_env("MNC", "70"),
  prometheus_metrics_port:
String.to_integer(System.get_env("PROMETHEUS_PORT", "9568")),
  heartbeat_interval:
String.to_integer(System.get_env("HEARTBEAT_INTERVAL", "10000"))

config :logger, :default_formatter,
  format: {OmniLogger.JsonFormatter, :format},
  metadata: :all
```

Parámetro	Tipo	Requerido	Predeterminado
`sbi_scheme`	cadena	No	`"http"`
`sbi_addr`	cadena	No	`"127.0.0.14"`
`sbi_port`	entero	No	`7777`
`nrf_uri`	cadena	No	`"http://127.0.0.1:777`

Parámetro	Tipo	Requerido	Predeterminado
`mcc`	cadena	No	`"999"`
`mnc`	cadena	No	`"70"`
`prometheus_metrics_port`	entero	No	`9568`

Parámetro	Tipo	Requerido	Predeterminado
`heartbeat_interval`	entero (ms)	No	`10000`

> Nota: el valor predeterminado compilado para `nrf_uri` en la configuración de tiempo de ejecución es `http://127.0.0.1:7777`, mientras que el valor predeterminado del accesor interno es `http://127.0.0.10:7777`. Cuando se aplica `runtime.exs`, el valor predeterminado efectivo es `http://127.0.0.1:7777`. Establezca `NRF_URI` explícitamente para evitar ambigüedades.

Comportamiento y Configuración de Notificaciones

Las siguientes claves de configuración de la aplicación `:omnissf` ajustan el comportamiento de selección de slice y notificación. Ninguna tiene una variable de entorno `runtime.exs`, por lo que se configuran a través de la configuración de la aplicación en lugar de una variable de entorno del sistema operativo.

Parámetro	Tipo	Requerido	Predetermin
`dynamic_nrf_discovery`	Booleano	No	`false`
`strict_availability`	Booleano	No	`false`

Parámetro	Tipo	Requerido	Predetermin
`nssf_notification_max_attempts`	Entero	No	`3`
`nssf_notification_backoff_ms`	Lista de enteros	No	`[200, 1000]`

Parámetro	Tipo	Requerido	Predetermir

Registrador

Parámetro	Tipo	Requerido	Predeterminado	I
`format`	tupla de formateador	No	`{OmniLogger.JsonFormatter, :format}`	-
`metadata`	átomo / lista	No	`:all`	-

El nivel de registro activo se puede cambiar en tiempo de ejecución sin reiniciar a través de la API OAM (`PATCH /oam/config/{id}` con `log_level`, o `POST /oam/log_level`). Los niveles válidos son `emergency`, `alert`, `critical`, `error`, `warning`, `warn`, `notice`, `info` y `debug`.

Configuración de Topología de Slice

La topología de slice se carga en el almacén `SliceConfig` al inicio desde las siguientes claves de configuración de la aplicación `:omnissf`. Cada una tiene un valor predeterminado incorporado que modela un único slice eMBB (SST 1, SD `0x000001`). Estas claves son estructurales (mapas y listas) y se configuran a través de la configuración de la aplicación en lugar de una sola variable de entorno; también se pueden cambiar en tiempo de ejecución a través de la API OAM (ver Gestión de Slice en Tiempo de Ejecución).

```
config :omnissf,
  nsi_list: [
    %{s_nssai: %{sst: 1, sd: "0x000001"}, nrf_uri:
"http://127.0.0.10:7777", nsi_id: "1"}
  ],
  allowed_nssai: %{
    "999-70" => [%{sst: 1, sd: "0x000001"}]
  },
  amf_set_mapping: %{
    "1-0x000001" => ["1"]
  },
  configured_nssai: %{}
```

`nsi_list` - Entradas de Instancia de Slice de Red

Cada entrada vincula un S-NSSAI al NRF que sirve el descubrimiento NF específico del slice para ese slice. El NSSF compara el S-NSSAI solicitado con esta lista; una coincidencia produce la información NSI devuelta al AMF, y ninguna coincidencia produce `403`.

Parámetro	Tipo	Requerido	Predeterminado	Var d Entor
`nsi_list`	lista de mapas	No	entrada eMBB única (abajo)	-
`s_nssai.sst`	entero	Sí	`1`	-
`s_nssai.sd`	cadena	No	`"0x000001"`	-
`nrf_uri`	cadena	Sí	`"http://127.0.0.10:7777"`	-
`nsi_id`	cadena	No	`"1"`	-

La coincidencia de SD es consciente de los comodines: un SD `nil` en la entrada configurada o en la solicitud coincide con cualquier SD, y los SD no nulos se comparan sin distinción de mayúsculas y minúsculas.

`allowed_nssai` - NSSAI Permitidos por PLMN

Asocia una clave PLMN `"mcc-mnc"` a la lista de S-NSSAIs permitidos para UEs de ese PLMN. Cuando un PLMN de consulta no tiene entrada, el NSSF vuelve al

PLMN de origen (`mcc-mnc` de la configuración de tiempo de ejecución). La lista resultante se devuelve como `allowedNssaiList` (tipo de acceso `3GPP_ACCESS`) y, cuando se proporciona un TAI, se filtra a los S-NSSAIs reportados disponibles en ese TAI.

Parámetro	Tipo	Requerido	Predeterminado	Var de Entorno	D
`allowed_nssai`	mapa	No	`{"999-70" => [...]}`	-	Ta N pe in Pl
clave del mapa	cadena `"mcc-mnc"`	Sí	`"999-70"`	-	Id Pl se co pe
`sst`	entero	Sí	`1`	-	S N pe
`sd`	cadena	No	`"0x000001"`	-	Sl N pe

`amf_set_mapping` - Conjunto AMF Objetivo por S-NSSAI

Asocia una clave S-NSSAI a los conjuntos AMF que pueden servir ese slice. Cuando está presente, el primer conjunto AMF se devuelve como `targetAmfSet`, indicando al AMF que redirija si no puede servir el slice por sí mismo.

Parámetro	Tipo	Requerido	Predeterminado	Var de Entorno
`amf_set_mapping`	mapa	No	`{"1-0x000001" => ["1"]}`	-
clave del mapa	cadena `"sst"` o `"sst-sd"`	Sí	`"1-0x000001"`	-
valor	lista de cadenas	Sí	`["1"]`	-

> La coincidencia de claves para `amf_set_mapping` y `configured_nssai` es **exacta**, a diferencia de `nsi_list`. La clave de búsqueda se construye a partir del S-NSSAI en la consulta: `"sst"` cuando el S-NSSAI solicitado no tiene SD, `"sst-sd"` cuando lo tiene. Una consulta que lleva un SD solo coincide con una clave `"sst-sd"`; una clave `"sst"` desnuda no la coincidirá, y viceversa. No hay comodines de SD aquí. Si sirves un slice con un SD, usa la forma `"sst-sd"` (por ejemplo, `"1-0x000001"`); si el S-NSSAI solicitado no tiene SD, usa la forma desnuda `"sst"` (por ejemplo, `"1"`).

`configured_nssai` - Mapeo de NSSAI Configurado (roaming)

Asocia una clave S-NSSAI a un valor de NSSAI Configurado devuelto al AMF para escenarios de roaming/mapeo. Vacío por defecto; cuando se establece, el valor se devuelve tal cual como `configuredNssai`. Esto se utiliza para dar a un UE en roaming el mapeo entre un S-NSSAI del PLMN que sirve (visitado) y el S-NSSAI correspondiente de su PLMN de origen, para que el UE pueda presentar un S-NSSAI de PLMN de origen cuando se mueva más tarde. El NSSF no

interpreta el valor: simplemente pasa el objeto que configures, por lo que debe tener la forma exacta que espera el AMF según TS 29.531.

Parámetro	Tipo	Requerido	Predeterminado	Var de Entorno
`configured_nssai`	mapa	No	`%{}` (vacío)	-
clave del mapa	cadena `"sst"` o `"sst-sd"`	No	(ninguna)	-
valor	objeto	No	(ninguno)	-

Ejemplo - devolver un mapeo de S-NSSAI de PLMN de origen para el slice de PLMN visitado `1-0x000001`:

```
config :omnissf,
  configured_nssai: %{
    "1-0x000001" => %{"sst" => 1, "sd" => "0x0000ff"}
  }
```

Deja `configured_nssai` vacío (`%{}`) para implementaciones no roaming; la respuesta omitirá entonces `configuredNssai` por completo.

Definiendo un slice de extremo a extremo

Un único slice no se describe mediante una clave de configuración: las cuatro claves anteriores son tablas independientes, y una respuesta de selección de slice se ensambla a partir de cualquiera de ellas que contenga una entrada coincidente. Para hacer que un nuevo slice sea completamente utilizable, normalmente se deben poblar dos, y opcionalmente las cuatro:

Clave	¿Requerido para que funcione el slice?	¿Qué controla en la respuesta?
`nsi_list`	**Sí**	Si el S-NSSAI se resuelve en absoluto. Una solicitud sin una entrada NSI coincidente se rechaza con `403 SNSSAI_NOT_FOUND`. Proporciona `nsiInformation` (NRF objetivo + ID NSI opcional)
`allowed_nssai`	**Sí**, por PLMN que sirve	Si se le dice al UE que puede usar el slice. Sin una entrada PLMN coincidente (y sin retroceso a PLMN de origen), la respuesta omite `allowedNssaiList`, por lo que el AMF no tiene NSSAI permitido para dar al UE
`amf_set_mapping`	Opcional	Agrega `targetAmfSet` para que el AMF pueda redirigir cuando no puede servir el slice por sí mismo. Omitir para implementaciones de conjunto AMF único
`configured_nssai`	Opcional (solo roaming)	Agrega `configuredNssai` para el mapeo de S-NSSAI de visitado a origen

Ejemplo trabajado - agregar slice SST `2` (URLLC), SD `0x000010`, servido por NRF `http://10.0.0.20:7777`, permitido para PLMN de origen `999-70`, redirigible al conjunto AMF `2`:

```
config :omnissf,
  nsi_list: [
    %{s_nssai: %{sst: 2, sd: "0x000010"}, nrf_uri:
"http://10.0.0.20:7777", nsi_id: "2"}
  ],
  allowed_nssai: %{
    "999-70" => [%{sst: 2, sd: "0x000010"}]
  },
  amf_set_mapping: %{
    "2-0x000010" => ["2"]
  }
```

Recuerda la asimetría de coincidencia de SD: `nsi_list` y `allowed_nssai` comparan SDs sin distinción de mayúsculas y minúsculas con `nil` actuando como un comodín, mientras que `amf_set_mapping` y `configured_nssai` están indexados por una cadena exacta `"sst-sd"`. Mantén la forma del SD consistente (por ejemplo, siempre `"0x000010"`) en las cuatro claves para que las claves se alineen.

El mismo slice también se puede agregar en tiempo de ejecución con `POST /oam/slice` (que escribe solo `nsi_list`); las tablas de NSSAI permitidos, conjuntos AMF y NSSAI configurados son solo de configuración y no son editables a través de la API OAM. Las adiciones de NSI en tiempo de ejecución se revierten al reiniciar.

Puertos de Gestión y Métricas

Los siguientes oyentes orientados al operador se inician junto con el endpoint SBI:

Oyente	Predeterminado	Var de Entorno	Descripción
API de Gestión / OAM	TCP `8443` (HTTPS)	-	API REST para estado, inspección de slice, estadísticas y acciones OAM en tiempo de ejecución
Métricas de Prometheus	TCP `9568`	`PROMETHEUS_PORT`	Endpoint de scraping de Prometheus que expone métricas de NSSF y BEAM

API de Gestión / OAM

Una API de gestión/OAM REST se sirve a través de HTTPS en el puerto `8443` (vinculado a `0.0.0.0`). Proporciona estado y estadísticas de solo lectura más controles OAM en tiempo de ejecución. Todas las respuestas son JSON, envueltas en un sobre estándar de `success` / `error`. Las rutas a continuación son relativas a la base de la API.

Estado e Inspección

Método	Ruta	Descripción
GET	`/status/api`	Vitalidad del servicio API
GET	`/status/nrf`	Estado de registro del NRF
GET	`/status/nf`	Estado del NF NSSF
GET	`/slice_config`	Instantánea de la configuración actual del slice
GET	`/api/slices`	S-NSSAIs configurados con mapeos NSI, conjuntos AMF objetivo, NSSAI permitidos por PLMN y mapeo de conjuntos AMF
GET	`/api/nssai_availability`	Disponibilidad de NSSAI por AMF actualmente almacenada
GET	`/api/statistics`	Estadísticas operativas de NSSF
GET	`/oam/config`	Configuración efectiva en tiempo de ejecución (dirección/puerto SBI, URI NRF, MCC/MNC, capacidad/prioridad NF, nivel de registro)

`GET /api/statistics` devuelve una instantánea de conteos útil para una rápida verificación de salud/capacidad sin scraping de Prometheus:

Campo	Significado
`total_slices`	Número de entradas NSI actualmente cargadas (tamaño de `nsi_list`)
`total_allowed_nssai_plmns`	Número de PLMNs con una entrada `allowed_nssai`
`total_amf_set_mappings`	Número de claves S-NSSAI en `amf_set_mapping`
`nssai_availability_count`	Número de AMFs que han reportado disponibilidad por TAI
`nssai_availability_subscription_count`	Número de suscripciones activas de cambio de disponibilidad

Gestión de Slice en Tiempo de Ejecución

Método	Ruta	Descripción
POST	`/oam/slice`	Agregar o actualizar un mapeo NSI. Cuerpo: `sst` (requerido), `nrf_uri` (requerido), `sd` opcional, `nsi_id`
DELETE	`/oam/slice/{id}`	Eliminar un mapeo NSI. `id` es la clave S-NSSAI `"sst"` o `"sst-sd"`. Devuelve `404` si no existe un NSI coincidente
DELETE	`/oam/nssai_availability/{id}`	Invalidar la disponibilidad de NSSAI almacenada para un AMF (`id` = `nfId`)

Los cambios de slice en tiempo de ejecución se aplican inmediatamente a las consultas de selección de slice posteriores, pero no se persisten; vuelven a la topología configurada al reiniciar.

Controles de Configuración en Tiempo de Ejecución

Método	Ruta	Descripción
PATCH	`/oam/config/{id}`	Actualizar la configuración en tiempo de ejecución. Claves permitidas: `nrf_uri`, `nf_capacity`, `nf_priority`, `log_level`. Claves/valores no válidos devuelven `400`
POST	`/oam/nrf/reregister`	Forzar el re-registro del NRF
POST	`/oam/log_level`	Cambiar el nivel de registro. Cuerpo: `level` (uno de `emergency`, `alert`, `critical`, `error`, `warning`, `warn`, `notice`, `info`, `debug`)

Para `PATCH /oam/config/{id}`, `nf_capacity` y `nf_priority` deben ser enteros y `log_level` debe ser una cadena de nivel válida; de lo contrario, la solicitud devuelve `400`.

← Resumen

Métricas de OmniNSSF

Las métricas se exponen en formato Prometheus en el puerto de métricas (por defecto `9568`, `PROMETHEUS_PORT`). Las métricas específicas de NSSF se enumeran a continuación; las métricas de BEAM VM/memoria (`beam.memory.*`, `beam.processes.count`, `beam.ports.count`, `beam.atom.count`, `beam.vm.uptime`) también se exportan.

Métricas de NSSF

Métrica	Tipo	Et
`omni_nssf_nsselection_requests_count`	Contador	`re` `nf`
`omni_nssf_ns_selection_requests_total`	Contador	`re`
`omni_nssf_nssai_availability_update_count`	Contador	`nf`
`omni_nssf_nssai_availability_delete_count`	Contador	`nf`
`omni_nssf_nssai_availability_subscribe_count`	Contador	(n
`omni_nssf_nssai_availability_unsubscribe_count`	Contador	(n
`omni_nssf_nssai_availability_updates_total`	Contador	`nf`
`omni_nssf_nssai_availability_patch_total`	Contador	`nf`
`omni_nssf_nssai_availability_patch_subscription_total`	Contador	(n

Métrica	Tipo	Et
`omni_nssf_nrf_registration_status`	Medidor	`nf`

> Existen dos contadores de selección de slice casi idénticos que no deben sumarse. `omni_nssf_nsselection_requests_count` utiliza valores de `result` `success`/`error` y también lleva `nf_type`, mientras que `omni_nssf_ns_selection_requests_total` utiliza valores de `result` `success`/`failure` y no tiene `nf_type`. Elija uno como su fuente de verdad para evitar contar doble las mismas solicitudes.

> `omni_nssf_nssai_availability_updates_total` (plural `updates`) y `omni_nssf_nssai_availability_update_count` (singular) están conectados al mismo evento de actualización PUT, por lo que se rastrean juntos. Cualquiera de los dos se puede usar para el volumen total de actualizaciones de disponibilidad.

Métricas de BEAM VM

Métrica	Tipo	Descripción
`beam_memory_total`	Medidor	Memoria total asignada por la VM (bytes)
`beam_memory_processes` / `beam_memory_processes_used`	Medidor	Memoria asignada a / utilizada por procesos (bytes)
`beam_memory_system`	Medidor	Memoria asignada al sistema VM (bytes)
`beam_memory_atom` / `beam_memory_atom_used`	Medidor	Memoria asignada a / utilizada por la tabla de átomos (bytes)
`beam_memory_binary`	Medidor	Memoria asignada a binarios (bytes)
`beam_memory_code`	Medidor	Memoria asignada a código cargado (bytes)
`beam_memory_ets`	Medidor	Memoria asignada a tablas ETS (bytes)
`beam_processes_count`	Medidor	Número de procesos activos
`beam_ports_count`	Medidor	Número de puertos abiertos
`beam_atom_count`	Medidor	Número de átomos en la tabla de átomos

Métrica	Tipo	Descripción
`beam_vm_uptime`	Medidor	Tiempo de actividad de la VM (segundos)

Ejemplo de PromQL

Tasa de error de selección de slice durante 5 minutos:

```
sum(rate(omni_nssf_nsselection_requests_count{result="error"}
[5m]))
  / sum(rate(omni_nssf_nsselection_requests_count[5m]))
```

Tasa de solicitudes de selección de slice por tipo de NF solicitado:

```
sum by (nf_type) (rate(omni_nssf_nsselection_requests_count[5m]))
```

Salud del registro NRF (alerta cuando no está registrado):

```
min(omni_nssf_nrf_registration_status) < 1
```

Tasa de actualización de disponibilidad de NSSAI por AMF:

```
sum by (nf_id)
(rate(omni_nssf_nssai_availability_update_count[5m]))
```

← Resumen

Guía de Operaciones de OmniNSSF

OmniNSSF implementa la Función de Selección de Cortes de Red (NSSF) del núcleo 5G. Resuelve consultas de selección de corte del AMF en una respuesta de selección de corte autorizada y recopila y autoriza la disponibilidad de cortes por Área de Seguimiento reportada por los AMFs.

La topología de cortes (mapeos de S-NSSAI a NRF, NSSAI permitidos por PLMN, mapeos de conjuntos de AMF y NSSAI configurados) se mantiene en un almacén `SliceConfig` en memoria que se carga desde la configuración de la aplicación al inicio y puede ajustarse en tiempo de ejecución a través de la API de gestión OAM. La disponibilidad de NSSAI por AMF y las suscripciones de disponibilidad se mantienen en un almacén `Context` en memoria. Este es un estado en tiempo de ejecución: se reconstruye a partir de la configuración y de los informes en vivo de AMF después de un reinicio.

OmniNSSF se registra con el NRF al inicio y mantiene su registro con latidos periódicos. Por defecto, los objetivos NRF específicos de corte son configurados por el operador en `nsi_list` y devueltos al AMF como información de NSI. El descubrimiento dinámico de NRF también está disponible como un comportamiento opcional: cuando `dynamic_nrf_discovery` está habilitado, en cada solicitud de selección de corte, el NSSF resuelve el NRF que sirve el corte en tiempo de ejecución desde el NRF raíz a través de Nnrf_NFDiscovery, volviendo a la `nsi_list` configurada estáticamente cuando el descubrimiento no es accesible o no devuelve nada. Esto está deshabilitado por defecto.

Rol de 3GPP y Referencias de Especificación

Especificación	Relevancia
TS 23.501 §5.15	Conceptos de corte de red - S-NSSAI, tipos de NSSAI (Solicitado / Permitido / Configurado / Suscrito), Instancia de Corte de Red, rol de NSSF
TS 23.502	Procedimientos - selección de corte durante el Registro y Establecimiento de Sesión PDU
TS 29.531	APIs de servicio Nnssf_NSSelection y Nnssf_NSSAIAvailability (HTTP/2 SBI)
TS 23.003	Formato S-NSSAI - SST (1 octeto) y SD opcional (3 octetos)

Conceptos de S-NSSAI y NSSAI

Una **Información de Asistencia para la Selección de Cortes de Red Única (S-NSSAI)** identifica un corte de red. Según TS 23.003, consiste en:

- **SST (Tipo de Corte/Servicio)** - obligatorio, un valor de 8 bits. Los valores estandarizados incluyen `1` = eMBB, `2` = URLLC, `3` = MIoT, `4` = V2X.
- **SD (Diferenciador de Corte)** - opcional, un valor de 24 bits que distingue múltiples cortes del mismo SST. Comúnmente escrito como una cadena de 6 dígitos hexadecimales como `0x000001`. Cuando SD está ausente, el S-NSSAI se aplica a cualquier/todos los diferenciadores de ese SST.

Un **NSSAI** es un conjunto de S-NSSAIs. El sistema 5G distingue varios tipos de NSSAI, todos los cuales el NSSF utiliza:

Tipo de NSSAI	Significado	Dónde lo utiliza OmniNSSF
NSSAI Solicitado	S-NSSAIs que el UE solicita usar, enviados por el AMF en la consulta de selección de corte	Entrada - extraído de la solicitud de información de corte de registro o sesión PDU
NSSAI Suscrito	S-NSSAIs a los que el UE tiene derecho según su suscripción	Entrada - respaldo cuando no hay NSSAI Solicitado presente en una consulta de registro
NSSAI Permitido	S-NSSAIs que el UE puede usar realmente en el PLMN/ área que sirve	Salida - calculado a partir del conjunto permitido por PLMN y filtrado por la disponibilidad de TAI
NSSAI Configurado	S-NSSAIs configurados para el UE en un PLMN (utilizados para itinerancia/mapeo)	Salida - devuelto cuando se define un mapeo de NSSAI Configurado para el S-NSSAI

El NSSF resuelve una consulta de la siguiente manera: coincide el S-NSSAI solicitado contra la lista de **Instancias de Corte de Red (NSI)** configurada para encontrar el NRF específico del corte, calcula el NSSAI Permitido para el PLMN que sirve (opcionalmente filtrado por la disponibilidad reportada de TAI) y agrega un conjunto de AMF objetivo y un mapeo de NSSAI Configurado cuando estos están configurados para el corte.

Puntos finales de SBI

Todos los puntos finales de SBI se sirven a través de la URL base `{sbi_scheme}://{sbi_addr}:{sbi_port}` (por defecto `http://127.0.0.14:7777`).

Nnssf_NSSelection (TS 29.531)

Método	Ruta	Descripción
GET	`/nnssf-nsselection/v2/network-slice-information`	Resolver la selección de corte para el AMF. Devuelve un objeto `AuthorizedNetworkSliceInfo`

Parámetros de consulta obligatorios: `nf-type`, `nf-id`. Un parámetro obligatorio faltante devuelve `400` con causa `MANDATORY_IE_MISSING`. La entrada del corte se proporciona a través de uno de `slice-info-request-for-pdu-session`, `slice-info-request-for-registration`, o un parámetro directo `snssai`. Los parámetros opcionales incluyen `tai`, `home-plmn-id`, y `supported-features`.

Códigos de respuesta:

Código	Significado
`200`	`AuthorizedNetworkSliceInfo` (información de NSI, y donde sea aplicable NSSAI Permitido, conjunto de AMF objetivo, NSSAI Configurado)
`400`	Parámetro obligatorio faltante o no se suministró información de corte
`403`	No hay NSI configurado para el S-NSSAI solicitado (causa `SNSSAI_NOT_FOUND`)

Nnssf_NSSAIAvailability (TS 29.531)

Método	Ruta	De
PUT	`/nnssf-nssaiavailability/v1/nssai-availability/{nfId}`	Almacenar la li soportados por Devuelve `AuthorizedNss`
PATCH	`/nnssf-nssaiavailability/v1/nssai-availability/{nfId}`	Actualizar parc disponibilidad JSON Patch (RF `application/j`
DELETE	`/nnssf-nssaiavailability/v1/nssai-availability/{nfId}`	Eliminar la disp almacenada pa `204`
POST	`/nnssf-nssaiavailability/v1/nssai-availability/subscriptions`	Crear una susc disponibilidad. un encabezado
PATCH	`/nnssf-nssaiavailability/v1/nssai-availability/subscriptions/{subscriptionId}`	Actualizar una JSON Patch (RF
DELETE	`/nnssf-nssaiavailability/v1/nssai-availability/subscriptions/{subscriptionId}`	Eliminar una su `204`

Un cuerpo de disponibilidad PUT/PATCH debe contener un array `supportedNssaiAvailabilityData` no vacío, cada entrada llevando un objeto `tai` y una `supportedSnssaiList` no vacía. Un PATCH sobre la disponibilidad almacenada (o sobre una suscripción) que haga referencia a un `{nfId}` / `{subscriptionId}` desconocido devuelve `404`. Un cuerpo de suscripción debe contener `nfNssaiAvailabilityUri`, una `taiList` no vacía, y un `event`. Cualquier ruta SBI no coincidente devuelve `404`.

Forma de los datos de disponibilidad y coincidencia de TAI

Cada entrada de `supportedNssaiAvailabilityData` tiene dos partes requeridas:

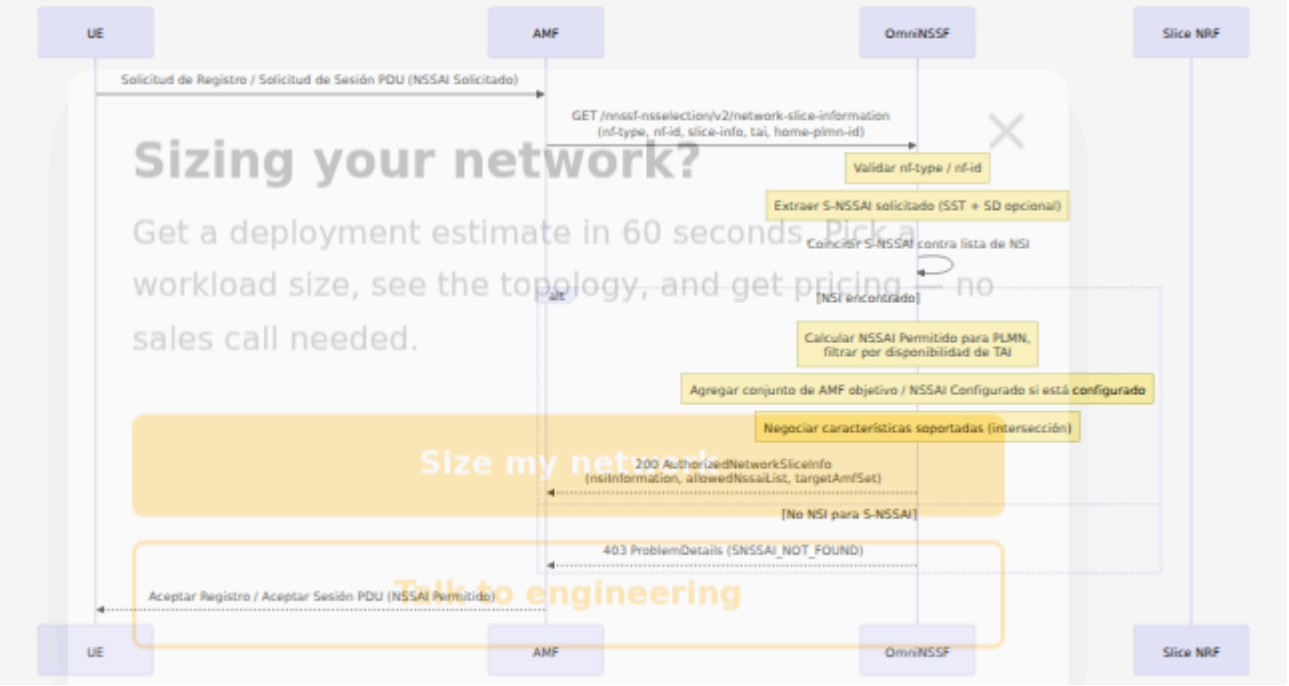
Campo	Forma	Significado
`tai`	`{ "plmnId": { "mcc", "mnc" }, "tac" }`	El Área de Seguimiento a la que se aplica la disponibilidad
`supportedSnssaiList`	array no vacío de `{ "sst", "sd"? }`	S-NSSAIs que el AMF reportante soporta en ese TAI

Cuando la selección de corte filtra más tarde un NSSAI Permitido por un TAI de consulta (ver Selección de Corte de Red), la coincidencia es **exacta en los tres de `mcc`, `mnc`, y `tac`**, y un S-NSSAI se considera disponible solo si alguna entrada reportada para ese TAI lo lista (SD comparado exactamente, o coincidiendo con cualquier SD cuando la consulta omite SD). Por lo tanto, un corte puede parecer no disponible simplemente porque el AMF reportó un TAI cuyo `tac` o PLMN difiere del que se encuentra en la consulta de selección: asegúrese de que el AMF reporte el mismo TAI que codifica en consultas posteriores.

Procedimientos Clave

Selección de Corte de Red (Nnssf_NSSelection)

Consultado por el AMF durante el Registro del UE y durante el Establecimiento de Sesión PDU. El NSSF resuelve el S-NSSAI solicitado en una respuesta de selección de corte autorizada.



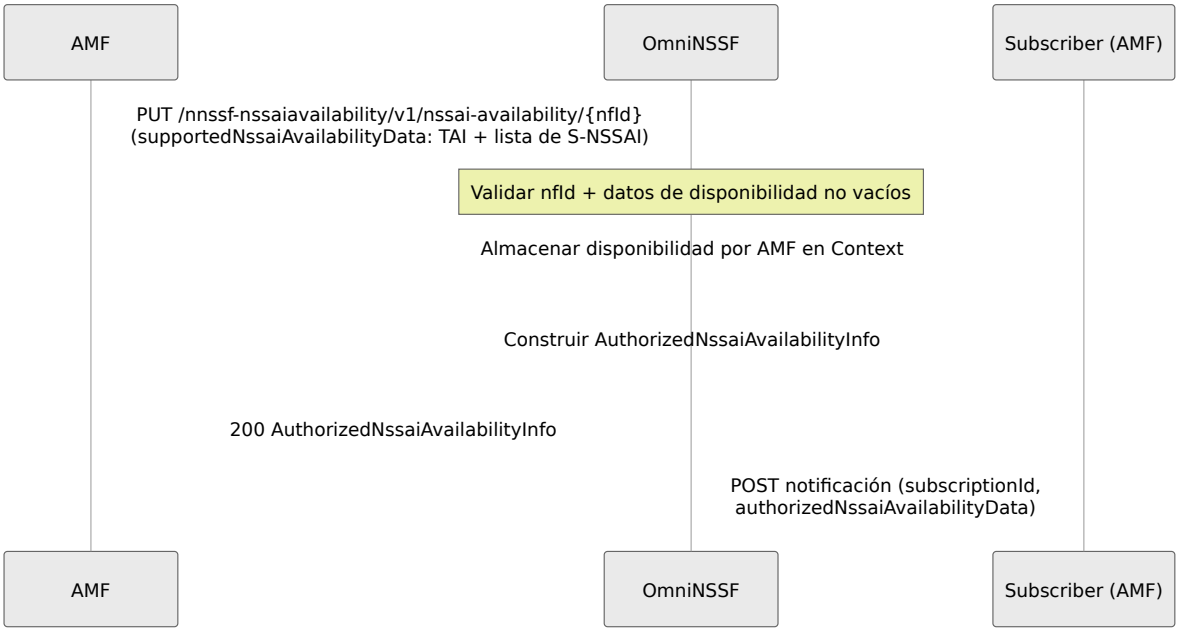
Sizing your network?
Get a deployment estimate in 60 seconds. Pick a workload size, see the topology, and get pricing — no sales call needed.
Size my network
Talk to engineering

Cuando se proporciona un TAI en la consulta y uno o más AMFs han reportado disponibilidad, el NSSF verifica que el S-NSSAI solicitado esté disponible en ese TAI y filtra el NSSAI Permitido en consecuencia. Si aún no se ha reportado disponibilidad, todos los cortes configurados se tratan como disponibles (predeterminado permisivo; establezca `strict_availability: true` para fallar cerrado). Cuando el AMF suministra un valor de `supported-features`, el NSSF devuelve la intersección negociada contra su propio conjunto de características (actualmente el bit de NssaiAvailability).

El NRF objetivo devuelto como `nsiInformation` normalmente proviene de la entrada coincidente de `nsi_list`. Cuando `dynamic_nrf_discovery` está habilitado, el NSSF resuelve en su lugar el NRF que sirve el corte en tiempo de ejecución consultando el NRF raíz a través de Nnrf_NFDiscovery para el S-NSSAI solicitado, y vuelve al NRF de `nsi_list` configurado cuando el descubrimiento está deshabilitado, no es accesible, o no devuelve nada. Consulte la Referencia de Configuración para la clave de configuración.

Actualización de Disponibilidad de NSSAI (Nnssf_NSSAIAvailability)

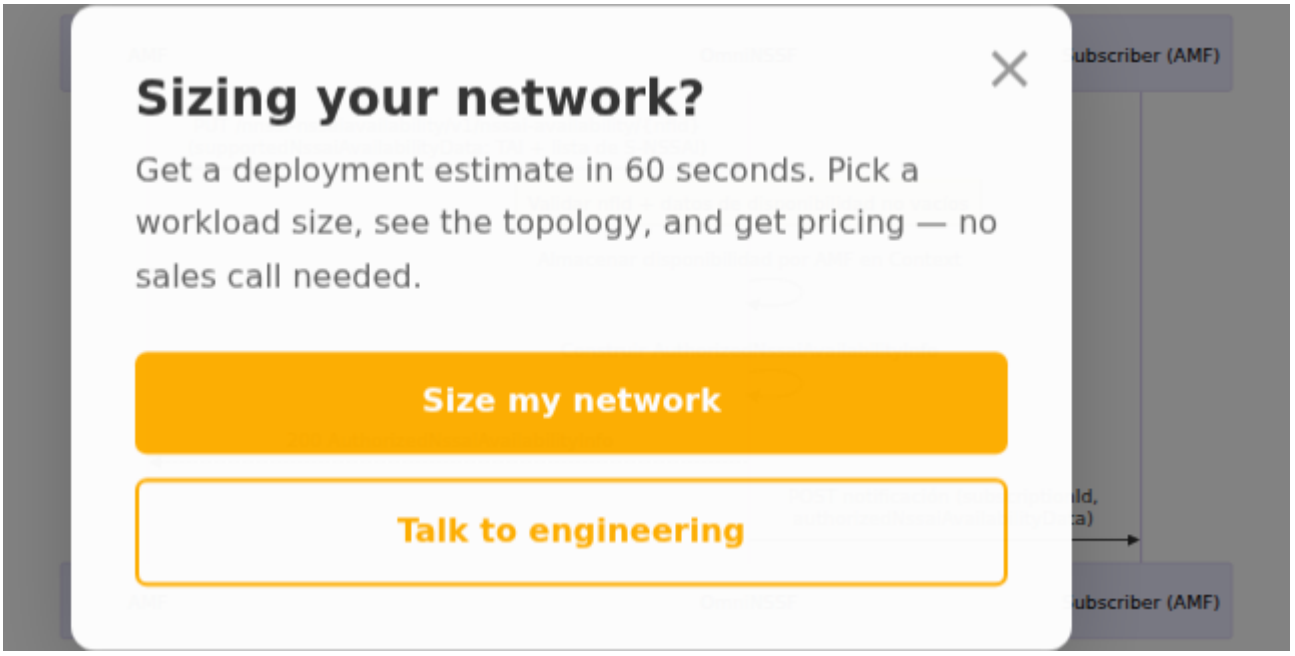
Los AMFs envían los S-NSSAIs que soportan por TAI. El NSSF almacena y autoriza esta información y notifica a cualquier suscriptor del cambio.



Un PATCH sobre la disponibilidad almacenada aplica las operaciones de JSON Patch (RFC 6902) al registro existente por AMF; un PATCH contra un `{nfId}` desconocido devuelve `404`. Tanto PUT como PATCH vuelven a notificar a los suscriptores con los datos de disponibilidad autorizados.

Suscripción y Notificación de Disponibilidad de NSSAI

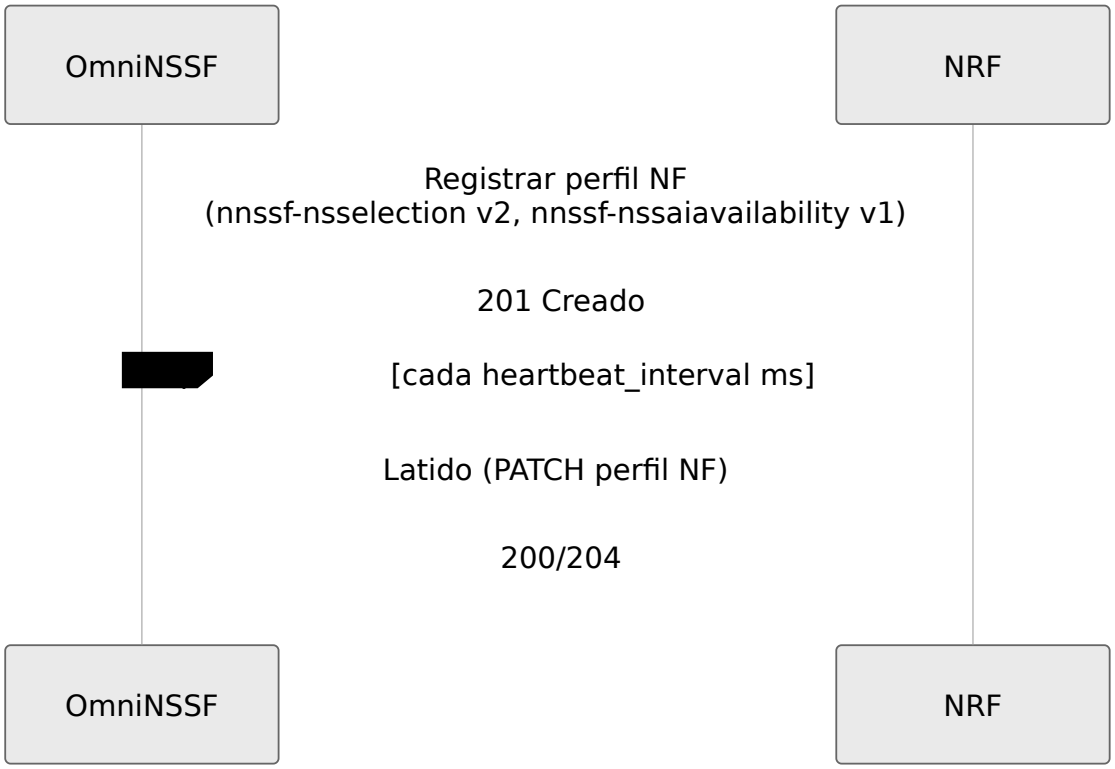
Los suscriptores registran una URI de callback; el NSSF envía una notificación de cambio a cada suscriptor cada vez que un PUT o PATCH actualiza la disponibilidad almacenada.



Las notificaciones son POSTs HTTP a la `nfNssaiAvailabilityUri` de cada suscriptor; la notificación lleva el `subscriptionId` y el `authorizedNssaiAvailabilityData`. Cada notificación se entrega fuera de la ruta de solicitud con reintentos limitados: hasta `nssf_notification_max_attempts` intentos (por defecto `3`) utilizando el horario de retroceso `nssf_notification_backoff_ms` (por defecto `200` ms luego `1000` ms), después de lo cual la notificación se descarta y se registra el fallo. Debido a que la entrega es fuera de la ruta de solicitud, un suscriptor lento o inaccesible no bloquea la actualización de disponibilidad que activó la notificación.

Registro y Latido de NRF

Al iniciar, el NSSF registra su perfil NF con el NRF, anunciando ambos servicios SBI, y luego se vuelve a registrar en un latido periódico.



El estado de registro se refleja en el medidor `omni_nssf_nrf_registration_status` y se puede forzar a volver a registrarse a través de la API OAM (`POST /oam/nrf/reregister`).

← Resumen

Solución de problemas de OmniNSSF

Problemas Comunes

Síntoma	Causa probable	
La selección de slice devuelve `400` `MANDATORY_IE_MISSING`	La consulta AMF carece de `nf-type` o `nf-id`	Confirme que parámetros d obligatorios
La selección de slice devuelve `400` "No se proporcionó información de slice"	La consulta no tiene ninguno de `slice-info-request-for-pdu-session`, `slice-info-request-for-registration` o `snssai`	Verifique que objeto slice-ir procedimient
La selección de slice devuelve `403` `SNSSAI_NOT_FOUND`	El S-NSSAI solicitado no tiene una entrada NSI coincidente	Agregue el S- (config) o a tr `/oam/slice`
La respuesta omite `allowedNssaiList`	No hay entrada `allowed_nssai` para el PLMN de la consulta y no hay coincidencia de respaldo de home-PLMN	Agregue el PL home) a `allo`
El slice parece no estar disponible en un TAI	Existe un informe de disponibilidad AMF en conflicto para ese TAI sin el S-NSSAI	Inspeccione `G` `/api/nssai_a` invalide los d `DELETE` `/oam/nssai_a`
No hay `targetAmfSet` en la respuesta	No hay entrada `amf_set_mapping` para la clave S-NSSAI	Agregue la cl `amf_set_mapp`
Estado de registro NRF `0`	NRF inalcanzable o `nrf_uri` incorrecto	Verifique `NRF_` registro con `P` `/oam/nrf/rer`

Síntoma	Causa probable	
PUT de disponibilidad rechazado `400`	Falta/ vacío `supportedNssaiAvailabilityData`, o una entrada carece de `tai` o de una `supportedSnssaiList` no vacía	Corrija la forn disponibilidac
PATCH de disponibilidad devuelve `404`	JSON Patch enviado para un `nfId` sin disponibilidad almacenada	Primero haga disponibilidac correcto
Creación de suscripción rechazada `400`	Cuerpo faltante `nfNssaiAvailabilityUri`, `taiList` o `event`	Proporcione lo suscripción re
PATCH/DELETE de suscripción devuelve `404`	`subscriptionId` desconocido	Confirme el `s` encabezado `l`
Suscriptores no notificados	`nfNssaiAvailabilityUri` de callback inalcanzable	Las notificaci con retroceso (`nssf_notifi` intentos, por ser descartad alcanzabilidad registros de N advertencia d