# Oracle® Communications Cloud Native Core, Network Slice Selection Function User Guide





Oracle Communications Cloud Native Core, Network Slice Selection Function User Guide, Release 25.1.100

G20523-02

Copyright © 2019, 2025, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

Documentation Accessibility	
Diversity and Inclusion	
Conventions	
Introduction	
1.1 Overview	
1.2 References	3
NSSF Supported Services	
2.1 Network Slice Selection Service	1
2.2 NSSAI Availability Service	5
NSSF Architecture	
NSSF Architecture	
NSSF Architecture  NSSF Supported Features	
NSSF Supported Features	
NSSF Supported Features 4.1 DNS SRV Based Selection of NRF in NSSF	6
NSSF Supported Features  4.1 DNS SRV Based Selection of NRF in NSSF  4.2 Deleting All Slices in a TAI Using PATCH Remove Operation	6
NSSF Supported Features  4.1 DNS SRV Based Selection of NRF in NSSF  4.2 Deleting All Slices in a TAI Using PATCH Remove Operation  4.3 Support for TLS	6 9 15
NSSF Supported Features  4.1 DNS SRV Based Selection of NRF in NSSF  4.2 Deleting All Slices in a TAI Using PATCH Remove Operation  4.3 Support for TLS  4.4 Traffic Segregation	6 9 15 17
NSSF Supported Features  4.1 DNS SRV Based Selection of NRF in NSSF  4.2 Deleting All Slices in a TAI Using PATCH Remove Operation  4.3 Support for TLS  4.4 Traffic Segregation  4.5 Support for Common Service APIs in CNC Console	6 9 15 17
NSSF Supported Features  4.1 DNS SRV Based Selection of NRF in NSSF  4.2 Deleting All Slices in a TAI Using PATCH Remove Operation  4.3 Support for TLS  4.4 Traffic Segregation  4.5 Support for Common Service APIs in CNC Console  4.6 Enhanced Computation of AllowedNSSAI in NSSF	15 17 17 19
NSSF Supported Features  4.1 DNS SRV Based Selection of NRF in NSSF 4.2 Deleting All Slices in a TAI Using PATCH Remove Operation 4.3 Support for TLS 4.4 Traffic Segregation 4.5 Support for Common Service APIs in CNC Console 4.6 Enhanced Computation of AllowedNSSAI in NSSF 4.7 LCI and OCI Headers	15 17 17 19 23
NSSF Supported Features  4.1 DNS SRV Based Selection of NRF in NSSF  4.2 Deleting All Slices in a TAI Using PATCH Remove Operation  4.3 Support for TLS  4.4 Traffic Segregation  4.5 Support for Common Service APIs in CNC Console  4.6 Enhanced Computation of AllowedNSSAI in NSSF  4.7 LCI and OCI Headers  4.8 Server Header in NSSF	15 17 17 19 23
NSSF Supported Features  4.1 DNS SRV Based Selection of NRF in NSSF 4.2 Deleting All Slices in a TAI Using PATCH Remove Operation 4.3 Support for TLS 4.4 Traffic Segregation 4.5 Support for Common Service APIs in CNC Console 4.6 Enhanced Computation of AllowedNSSAI in NSSF 4.7 LCI and OCI Headers 4.8 Server Header in NSSF 4.9 Support for User-Agent Header	15 17 17 19 23 26 28
NSSF Supported Features  4.1 DNS SRV Based Selection of NRF in NSSF  4.2 Deleting All Slices in a TAI Using PATCH Remove Operation  4.3 Support for TLS  4.4 Traffic Segregation  4.5 Support for Common Service APIs in CNC Console  4.6 Enhanced Computation of AllowedNSSAI in NSSF  4.7 LCI and OCI Headers  4.8 Server Header in NSSF  4.9 Support for User-Agent Header  4.10 Ingress Gateway Pod Protection	1 6 9 15 17 17 19 23 26 28 34 38
NSSF Supported Features  4.1 DNS SRV Based Selection of NRF in NSSF  4.2 Deleting All Slices in a TAI Using PATCH Remove Operation  4.3 Support for TLS  4.4 Traffic Segregation  4.5 Support for Common Service APIs in CNC Console  4.6 Enhanced Computation of AllowedNSSAI in NSSF  4.7 LCI and OCI Headers  4.8 Server Header in NSSF  4.9 Support for User-Agent Header  4.10 Ingress Gateway Pod Protection  4.11 Monitoring the Availability of SCPs using SCP Health APIs	26 28 28 28 34

4.14	Deleting Subscription on 404 SUBSCRIPITON_NOT_FOUND Response from AMF	39
4.15	DNS SRV Based Selection of SCP in NSSF	46
4.16	OAuth Access Token Based Authorization	50
4.17	Overload Control based on Percentage Discards	57
4.18	Autopopulation of Configuration Using NRF Content	61
4.19	Auto-Population of Configuration Based on NSAvailability Update	63
4.20	Handover from EPS to 5G	70
4.21	Feature Negotiation	76
4.22	Subscription Modification Feature	79
4.23	Empty Authorized NSSAI Availability Notification	81
4.24	Optimized NSSAI Availability Data Encoding and TAI Range	84
4.25	Georedundancy	89
4.26	Time of the Day Based Network Slice Instance Selection	94
4.27	Multiple PLMN Support	97
4.28	Support Indirect Communication	100
4.29	IPv6 Support	108
4.30	Supports Integration with ASM	110
4.31	Supports Compression Using Accept-Encoding or Content-Encoding gzip	110
4.32	Dynamic Log Level Update	112
4.33	NF Authentication using TLS Certificate	113
	Distriction from Distributed Daniel of Comics (DDoC) Attack through Data Limiting	115
4.34	Protection from Distributed Denial-of-Service (DDoS) Attack through Rate Limiting	113
4.34 4.35	Automated Testing Suite Support	118
4.35	Automated Testing Suite Support	
4.35	, , ,	
4.35 Conf	Automated Testing Suite Support	
4.35 Conf	Automated Testing Suite Support	118
4.35 Conf 5.1 5.2	Automated Testing Suite Support  iguring NSSF using CNC Console  Support for Multicluster Deployment	118
4.35 Conf 5.1 5.2 5.3	Automated Testing Suite Support  iguring NSSF using CNC Console  Support for Multicluster Deployment  CNC Console Interface	118 1 1
4.35 Conf 5.1 5.2 5.3 5.3	Automated Testing Suite Support  Figuring NSSF using CNC Console  Support for Multicluster Deployment  CNC Console Interface  NSSF Configuration	118 1 1 2
4.35 Conf 5.1 5.2 5.3 5.3	Automated Testing Suite Support  iguring NSSF using CNC Console  Support for Multicluster Deployment CNC Console Interface  NSSF Configuration 3.1 AMF Set	118 1 1 2 3
4.35  Conf 5.1 5.2 5.3 5.6 5.6	Automated Testing Suite Support  Figuring NSSF using CNC Console  Support for Multicluster Deployment  CNC Console Interface  NSSF Configuration  3.1 AMF Set  3.2 AMF Resolution	118 1 1 2 3 4
4.35  Conf 5.1 5.2 5.3 5.3 5.5 5.5	Automated Testing Suite Support  iguring NSSF using CNC Console  Support for Multicluster Deployment CNC Console Interface NSSF Configuration 3.1 AMF Set 3.2 AMF Resolution 3.3 Configured SNSSAI	118 1 1 2 3 4 6
4.35  Conf 5.1 5.2 5.3 5.3 5.5 5.5	Automated Testing Suite Support  Figuring NSSF using CNC Console  Support for Multicluster Deployment  CNC Console Interface  NSSF Configuration  3.1 AMF Set  3.2 AMF Resolution  3.3 Configured SNSSAI  3.4 Georedundant Sites	118 1 1 2 3 4 6 6
5.1 5.2 5.3 5.3 5.3 5.3 5.3	Automated Testing Suite Support  Figuring NSSF using CNC Console  Support for Multicluster Deployment CNC Console Interface  NSSF Configuration 3.1 AMF Set 3.2 AMF Resolution 3.3 Configured SNSSAI 3.4 Georedundant Sites 3.5 NSI Profile	118 1 1 2 3 4 6 6 7
4.35  Conf 5.1 5.2 5.3 5.3 5.5 5.5 5.5 5.6	Automated Testing Suite Support  Figuring NSSF using CNC Console  Support for Multicluster Deployment  CNC Console Interface  NSSF Configuration  3.1 AMF Set  3.2 AMF Resolution  3.3 Configured SNSSAI  3.4 Georedundant Sites  3.5 NSI Profile  3.6 NSSAI Auth	118 1 1 2 3 4 6 7 8
4.35  Conf 5.1 5.2 5.3 5.6 5.6 5.6 5.6 5.6	Automated Testing Suite Support  Figuring NSSF using CNC Console  Support for Multicluster Deployment  CNC Console Interface  NSSF Configuration  3.1 AMF Set  3.2 AMF Resolution  3.3 Configured SNSSAI  3.4 Georedundant Sites  3.5 NSI Profile  3.6 NSSAI Auth  3.7 NSS Rule	118 1 1 2 3 4 6 6 7 8 9
4.35  Conf 5.1 5.2 5.3 5.3 5.3 5.5 5.5 5.5 5.5	Automated Testing Suite Support  Figuring NSSF using CNC Console  Support for Multicluster Deployment CNC Console Interface NSSF Configuration 3.1 AMF Set 3.2 AMF Resolution 3.3 Configured SNSSAI 3.4 Georedundant Sites 3.5 NSI Profile 3.6 NSSAI Auth 3.7 NSS Rule 3.8 Time Profile	118 1 1 2 3 4 6 6 7 8 9 11
4.35  Conf 5.1 5.2 5.3 5.6 5.6 5.6 5.6 5.6 5.6 5.6 5.6	Automated Testing Suite Support  Figuring NSSF using CNC Console  Support for Multicluster Deployment CNC Console Interface NSSF Configuration 3.1 AMF Set 3.2 AMF Resolution 3.3 Configured SNSSAI 3.4 Georedundant Sites 3.5 NSI Profile 3.6 NSSAI Auth 3.7 NSS Rule 3.8 Time Profile 3.9 Logging Level Options	118 1 1 2 3 4 6 6 7 8 9 11 12
4.35  Conf 5.1 5.2 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3	Automated Testing Suite Support  Figuring NSSF using CNC Console  Support for Multicluster Deployment CNC Console Interface NSSF Configuration 3.1 AMF Set 3.2 AMF Resolution 3.3 Configured SNSSAI 3.4 Georedundant Sites 3.5 NSI Profile 3.6 NSSAI Auth 3.7 NSS Rule 3.8 Time Profile 3.9 Logging Level Options 3.10 PLMN Level NSI Profiles	118 1 1 2 3 4 6 6 7 8 9 11 12 13
4.35  Conf 5.1 5.2 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3	Automated Testing Suite Support  Figuring NSSF using CNC Console  Support for Multicluster Deployment CNC Console Interface  NSSF Configuration 3.1 AMF Set 3.2 AMF Resolution 3.3 Configured SNSSAI 3.4 Georedundant Sites 3.5 NSI Profile 3.6 NSSAI Auth 3.7 NSS Rule 3.8 Time Profile 3.9 Logging Level Options 3.10 PLMN Level NSI Profiles 3.11 Mapping of Nssai	118 1 1 2 3 4 6 6 7 8 9 11 12 13 14
	4.16 4.17 4.18 4.19 4.20 4.21 4.22 4.23 4.24 4.25 4.26 4.27 4.28 4.29 4.30 4.31 4.32	<ul> <li>4.16 OAuth Access Token Based Authorization</li> <li>4.17 Overload Control based on Percentage Discards</li> <li>4.18 Autopopulation of Configuration Using NRF Content</li> <li>4.19 Auto-Population of Configuration Based on NSAvailability Update</li> <li>4.20 Handover from EPS to 5G</li> <li>4.21 Feature Negotiation</li> <li>4.22 Subscription Modification Feature</li> <li>4.23 Empty Authorized NSSAI Availability Notification</li> <li>4.24 Optimized NSSAI Availability Data Encoding and TAI Range</li> <li>4.25 Georedundancy</li> <li>4.26 Time of the Day Based Network Slice Instance Selection</li> <li>4.27 Multiple PLMN Support</li> <li>4.28 Support Indirect Communication</li> <li>4.29 IPv6 Support</li> <li>4.30 Supports Integration with ASM</li> <li>4.31 Supports Compression Using Accept-Encoding or Content-Encoding gzip</li> <li>4.32 Dynamic Log Level Update</li> </ul>

	5.4 Common S	Services Configuration	19
	5.4.1 Egre	ess Gateway	19
	5.4.1.1	Peer Configuration	19
	5.4.1.2	Peer Set Configuration	20
	5.4.1.3	Peer Monitoring Configuration	21
	5.4.1.4	Routes Configuration	22
	5.4.1.5	SBI Error Action Sets	24
	5.4.1.6	SBI Error Criteria Sets	24
	5.4.1.7	User Agent Header Generation	25
	5.4.2 Ingre	ess Gateway Configuration	26
	5.4.2.1	Error Code Profiles	26
	5.4.2.2	Create Overload Control Discard Policies	27
	5.4.2.3	Discard Policy Mapping	28
	5.4.2.4	Error Code Series	29
	5.4.2.5	Routes Configuration	30
	5.4.2.6	OAuth Validator Configurations	31
	5.4.2.7	Server Header Details	32
	5.4.2.8	Pod Protection	33
	5.5 cnDBTier	APIs	36
6		cs, KPIs, and Alerts	
6	6.1 NSSF Met	trics	1
6	6.1 NSSF Met 6.1.1 NSS	trics SF Success Metrics	4
6	6.1 NSSF Met 6.1.1 NSS 6.1.2 NSS	trics SF Success Metrics SF Error Metrics	4 11
6	6.1 NSSF Met 6.1.1 NSS 6.1.2 NSS 6.1.3 NSS	trics 6F Success Metrics 6F Error Metrics 6F Common metrics	4 11 17
6	6.1 NSSF Met 6.1.1 NSS 6.1.2 NSS 6.1.3 NSS 6.1.4 NSS	trics 6F Success Metrics 6F Error Metrics 6F Common metrics 6F OAuth Metrics	4 11 17 22
6	6.1 NSSF Met 6.1.1 NSS 6.1.2 NSS 6.1.3 NSS 6.1.4 NSS 6.1.5 Man	trics 6F Success Metrics 6F Error Metrics 6F Common metrics 6F OAuth Metrics 10 aged Objects Metrics	4 11 17 22 25
6	6.1 NSSF Met 6.1.1 NSS 6.1.2 NSS 6.1.3 NSS 6.1.4 NSS 6.1.5 Man 6.1.6 Perf-	trics 6F Success Metrics 6F Error Metrics 6F Common metrics 6F OAuth Metrics 6 aged Objects Metrics 6-info metrics for Overload Control	4 11 17 22
6	6.1 NSSF Met 6.1.1 NSS 6.1.2 NSS 6.1.3 NSS 6.1.4 NSS 6.1.5 Man 6.1.6 Perf- 6.1.7 Egre	trics 6F Success Metrics 6F Error Metrics 6F Common metrics 6F OAuth Metrics 6aged Objects Metrics 6-info metrics for Overload Control 6ass Gateway Metrics	4 11 17 22 25 36 37
6	6.1 NSSF Met 6.1.1 NSS 6.1.2 NSS 6.1.3 NSS 6.1.4 NSS 6.1.5 Man 6.1.6 Perf- 6.1.7 Egre	trics 6F Success Metrics 6F Error Metrics 6F Common metrics 6F OAuth Metrics 6F aged Objects Metrics 6F-info metrics for Overload Control 6F-info metrics for Overload Control 6F-info Metrics	4 11 17 22 25 36
6	6.1 NSSF Met 6.1.1 NSS 6.1.2 NSS 6.1.3 NSS 6.1.4 NSS 6.1.5 Man 6.1.6 Perf- 6.1.7 Egre 6.1.8 Ingre 6.2 NSSF KPI	trics 6F Success Metrics 6F Error Metrics 6F Common metrics 6F OAuth Metrics 6F aged Objects Metrics 6F-info metrics for Overload Control 6F-info metrics for Overload Control 6F-info Metrics	4 11 17 22 25 36 37 40
6	6.1 NSSF Met 6.1.1 NSS 6.1.2 NSS 6.1.3 NSS 6.1.4 NSS 6.1.5 Man 6.1.6 Perf- 6.1.7 Egre 6.1.8 Ingre 6.2 NSSF KPI 6.2.1 NSS	trics 6F Success Metrics 6F Error Metrics 6F Common metrics 6F OAuth Metrics 6 aged Objects Metrics 6 info metrics for Overload Control 6 ess Gateway Metrics 6 ess Gateway Metrics 6 stateway Metrics	40 11 17 22 25 36 37 40 41
6	6.1 NSSF Met 6.1.1 NSS 6.1.2 NSS 6.1.3 NSS 6.1.4 NSS 6.1.5 Man 6.1.6 Perf- 6.1.7 Egre 6.1.8 Ingre 6.2 NSSF KPI 6.2.1 NSS 6.2.2 NSA	trics SF Success Metrics SF Error Metrics SF Common metrics SF OAuth Metrics laged Objects Metrics sinfo metrics for Overload Control less Gateway Metrics less Gateway Metrics Selection KPIs	4 11 17 22 25 36 37 40 41
6	6.1 NSSF Met 6.1.1 NSS 6.1.2 NSS 6.1.3 NSS 6.1.4 NSS 6.1.5 Man 6.1.6 Perf- 6.1.7 Egre 6.1.8 Ingre 6.2 NSSF KPI 6.2.1 NSS 6.2.2 NSA	trics 6F Success Metrics 6F Error Metrics 6F Common metrics 6F OAuth Metrics 1 aged Objects Metrics 1 info metrics for Overload Control 1 ess Gateway Metrics 1 ess Gateway Metrics 1 s 1 Selection KPIs 1 Availability KPIs 1 ess Gateway KPIs	4 11 17 22 25 36 37 40 41 41 42
6	6.1 NSSF Met 6.1.1 NSS 6.1.2 NSS 6.1.3 NSS 6.1.4 NSS 6.1.5 Man 6.1.6 Perf- 6.1.7 Egre 6.1.8 Ingre 6.2 NSSF KPI 6.2.1 NSS 6.2.2 NSA 6.2.3 Ingre 6.3 NSSF Aler	trics 6F Success Metrics 6F Error Metrics 6F Common metrics 6F OAuth Metrics 1 aged Objects Metrics 1 info metrics for Overload Control 1 ess Gateway Metrics 1 ess Gateway Metrics 1 s 1 Selection KPIs 1 Availability KPIs 1 ess Gateway KPIs	4 11 17 22 25 36 37 40 41 41 42 43
6	6.1 NSSF Met 6.1.1 NSS 6.1.2 NSS 6.1.3 NSS 6.1.4 NSS 6.1.5 Man 6.1.6 Perf- 6.1.7 Egre 6.1.8 Ingre 6.2 NSSF KPI 6.2.1 NSS 6.2.2 NSA 6.2.3 Ingre 6.3 NSSF Aler	trics 6F Success Metrics 6F Error Metrics 6F Common metrics 6F OAuth Metrics 6 aged Objects Metrics 6 info metrics for Overload Control 6 ess Gateway Metrics 6 sess Gateway Metrics 7 sess Gateway Metrics 8 sess Gateway Metrics 8 sess Gateway Metrics 8 sess Gateway KPIs	4 11 17 22 25 36 37 40 41 41 42 43
6	6.1 NSSF Met 6.1.1 NSS 6.1.2 NSS 6.1.3 NSS 6.1.4 NSS 6.1.5 Man 6.1.6 Perf- 6.1.7 Egre 6.1.8 Ingre 6.2 NSSF KPI 6.2.1 NSS 6.2.2 NSA 6.2.3 Ingre 6.3 NSSF Aler 6.3.1 Syst	trics 6F Success Metrics 6F Error Metrics 6F Common metrics 6F OAuth Metrics 6F OAuth Metrics 6F aged Objects Metrics 6F-info metrics for Overload Control 6F-i	4 11 17 22 25 36 37 40 41 41 42 43 44
6	6.1 NSSF Met 6.1.1 NSS 6.1.2 NSS 6.1.3 NSS 6.1.4 NSS 6.1.5 Man 6.1.6 Perf- 6.1.7 Egre 6.1.8 Ingre 6.1.8 Ingre 6.2 NSSF KPI 6.2.1 NSS 6.2.2 NSA 6.2.3 Ingre 6.3 NSSF Aler 6.3.1 Syst 6.3.1.1	trics 6F Success Metrics 6F Error Metrics 6F Common metrics 6F OAuth Metrics 6 aged Objects Metrics 6 info metrics for Overload Control 6 ass Gateway Metrics 6 sess Gateway Metrics 7 sess Gateway Metrics 8 sess Gateway Metrics 8 sess Gateway Metrics 9 sess Gateway Metrics 18 sess Gateway KPIs	4 11 17 22 25 36 37 40 41 41 42 43 44 45
6	6.1 NSSF Met 6.1.1 NSS 6.1.2 NSS 6.1.3 NSS 6.1.4 NSS 6.1.5 Man 6.1.6 Perf- 6.1.7 Egre 6.1.8 Ingre 6.2 NSSF KPI 6.2.1 NSS 6.2.2 NSA 6.2.2 NSA 6.2.3 Ingre 6.3 NSSF Aler 6.3.1 Syst 6.3.1.1 6.3.1.2	trics SF Success Metrics SF Error Metrics SF Common metrics SF OAuth Metrics Alaged Objects Metrics Finfo metrics for Overload Control Bess Gateway Metrics Bess Gateway KPIs Bess Gateway KPIs Bess Gateway KPIs Bets Gateway KPIs	41 11 17 22 25 36 37 40 41 41 42 43 44 45 45

5.3.15 Trusted Amf

18

6.3	3.1.5 OcnssfAvailabilitySer	viceDown 49
6.3	3.1.6 OcnssfConfigurationS	serviceDown 50
6.3	3.1.7 OcnssfAppInfoService	eDown 51
6.3	3.1.8 OcnssflngressGatewa	ayServiceDown 52
6.3	3.1.9 OcnssfEgressGatewa	yServiceDown 53
6.3	3.1.10 OcnssfOcpmConfigS	ServiceDown 54
6.3	3.1.11 OcnssfPerfInfoServio	ceDown 55
6.3	3.1.12 OcnssfNrfClientMan	agementServiceDown 56
6.3	3.1.13 OcnssfNrfClientDisc	overyServiceDown 57
6.3	3.1.14 OcnssfAlternateRou	teServiceDown 58
6.3	3.1.15 OcnssfAuditorServic	eDown 59
6.3	3.1.16 OcnssfTotalIngressT	rafficRateAboveMinorThreshold 60
6.3	3.1.17 OcnssfTotalIngressT	rafficRateAboveMajorThreshold 61
6.3	3.1.18 OcnssfTotalIngressT	rafficRateAboveCriticalThreshold 62
6.3	3.1.19 OcnssfTransactionE	rrorRateAbove0.1Percent 62
6.3	3.1.20 OcnssfTransactionE	rrorRateAbove1Percent 63
6.3	3.1.21 OcnssfTransactionE	rrorRateAbove10Percent 64
6.3	3.1.22 OcnssfTransactionE	rrorRateAbove25Percent 64
6.3	3.1.23 OcnssfTransactionE	rrorRateAbove50Percent 65
6.3	3.1.24 OcnssfIngressGatev	vayPodCongestionStateWarning 66
6.3	3.1.25 OcnssfIngressGatev	vayPodCongestionStateMajor 67
6.3	3.1.26 OcnssfIngressGatev	vayPodResourceStateWarning 67
6.3	3.1.27 OcnssfIngressGatev	vayPodResourceStateMajor 68
6.3.2	Application Level Alerts	68
6.3	3.2.1 ocnssfPolicyNotFound	dWarning 69
6.3	3.2.2 ocnssfPolicyNotFound	dMajor 69
6.3	3.2.3 ocnssfPolicyNotFound	dCritical 70
6.3	3.2.4 OcnssfOverloadThres	holdBreachedL1 71
6.3	3.2.5 OcnssfOverloadThres	holdBreachedL2 71
6.3	3.2.6 OcnssfOverloadThres	sholdBreachedL3 72
6.3	3.2.7 OcnssfOverloadThres	sholdBreachedL4 73
6.3	3.2.8 OcnssfScpMarkedAst	Jnavailable 73
6.3	3.2.9 OcnssfAllScpMarked	AsUnavailable 74
6.3	3.2.10 OcnssfTLSCertificate	eExpireMinor 74
6.3	3.2.11 OcnssfTLSCertificate	eExpireMajor 74
6.3	3.2.12 OcnssfTLSCertificate	eExpireCritical 75
6.3	3.2.13 OcnssfNrfInstancesI	nDownStateMajor 75
6.3	3.2.14 OcnssfAllNrfInstance	esInDownStateCritical 76
6.3.3	NSSF Alert Configuration	76
6.3.4	Configuring SNMP Notifier	81



# **Preface**

- Documentation Accessibility
- Diversity and Inclusion
- Conventions

# **Documentation Accessibility**

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <a href="http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc">http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc</a>.

#### **Access to Oracle Support**

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

# **Diversity and Inclusion**

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

## Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

# My Oracle Support

My Oracle Support (<a href="https://support.oracle.com">https://support.oracle.com</a>) is your initial point of contact for all product support and training needs. A representative at Customer Access Support can assist you with My Oracle Support registration.

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at <a href="http://www.oracle.com/us/support/contact/index.html">http://www.oracle.com/us/support/contact/index.html</a>. When calling, make the selections in the sequence shown below on the Support telephone menu:

- 1. Select 2 for New Service Request.
- 2. Select **3** for Hardware, Networking and Solaris Operating System Support.
- 3. Select one of the following options:
  - For Technical issues such as creating a new Service Request (SR), select 1.
  - For Non-technical issues such as registration or assistance with My Oracle Support, select **2**.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

# Acronyms

The following table provides information about the acronyms used in the document:

Table 1 Acronyms

Field	Description
3GPP	3rd Generation Partnership Project
5GC	5G Core Network
5GS	5G System
Subscribed S-NSSAI	5G uses this as a default when the UE does not send a Requested NSSAI
AMF	Access and Mobility Management Function
API	Application Programming Interface
ASM	Aspen Service Mesh
CA	Certificate Authority
CNC	Cloud Native Core
CLI	Command Line Interface
CN	Common Name
CSP	Communication Service Provider
OKE	Container Engine for Kubernetes
DNN	Data Network Name
DB	Database
EGW	Egress Gateway
EANAN	Empty Authorized NSSAI Availability Notification
eMBB	enhanced Mobile Broadband
EPC	Evolved Packet Core. It is a framework for providing converged voice and data on a 4G Long-Term Evolution (LTE) network.
EPS	Evolved Packet System. It is a Mobility Management (EMM) protocol that provides procedures for the control of mobility when the User Equipment (UE) uses the Evolved UMTS Terrestrial Radio Access Network (E-UTRAN). EPS is a combination of E-UTRAN, EPC and UE.
FQDN	Fully Qualified Domain Name
GR	Georedundant
HNS	Hierarchical Namespace
H-NSSF	Home NSSF
HPLMN	Home Public Land Mobile Network
HTTPS	Hypertext Transfer Protocol Secure
IE	Information Element
KPI	Key Performance Indicator
MIoT	Massive Internet of Things
MPS	Messages Per Second
MOS	My Oracle Support
NWDAF	Network Data Analytics Function
NDB	Network Data Broker
NF	Network Function
NSI	Network Slice Instance



#### Table 1 (Cont.) Acronyms

Field	Description
NSI ID	Network Slice Instance Identifier
NSSAI	Network Slice Selection Assistance Information
NS	Network Slice. A logical network that provides specific network capabilities and network characteristics.
NSS	Network Switching Subsystem
Allowed NSSAI	NSSAI provided by the serving PLMN during a registration procedure, indicating the S-NSSAIs values the UE could use in the serving PLMN for the current registration area.
Requested NSSAI	NSSAI provided by the UE to the serving PLMN during registration.
Configured NSSAI	NSSAI provisioned in the UE applicable to one or more PLMNs.
ONSSAI	Optimized NSSAI Availability Data Encoding feature
OCI	Oracle Cloud Infrastructure
CNE	Oracle Communications Cloud Native Core, Cloud Native Environment
NRF	Oracle Communications Cloud Native Core, Network Repository Function
NSSF	Oracle Communications Cloud Native Core, Network Slice Selection Function
SEPP	Oracle Communications Cloud Native Core, Security Edge Protection Proxy
SCP	Oracle Communications Cloud Native Core, Service Communication Proxy
UDR	Oracle Communications Cloud Native Core, Unified Data Repository
OHC	Oracle Help Center
OSDC	Oracle Service Delivery Cloud
PEI	Permanent Equipment Identifier
PDU	Protocol Data Unit
PLMN	Public Land Mobile Network
RAN	Radio Access Network
SBA	Service Based Architecture
SBI	Service Based Interface
SVC	Services
SSC	Session and Service Continuity
SMF	Session Management Function
S-NSSAI	Single Network Slice Selection Assistance Information
SD	Slice Differentiator
SST	Slice or Service type
SUMOD	Subscription Modification
SUPI	Subscription Permanent Identifier
Restricted S-NSSAI	This is an information element (IE) that contains restricted S-NSSAI(s) per PLMN for a Tracking Area(TA). If the restricted SNssai is not present, no restricted S-NSSAI is applicable to the TA. If present, this IE (restrictedSnssai) is included only by the NSSF
TA	Tracking Area
TAC	Tracking Area Code
TAI	Tracking Area Identifier
TLS	Transport Layer Security
URLLC	Ultra-Reliable Low Latency Communications
UDM	Unified Data Management
URI	Uniform Resource Identifier



#### Table 1 (Cont.) Acronyms

Field	Description
UE	User Equipment
V-NSSF	Visited NSSF
VPLMN	Visited Public Land Mobile Network

# What's New in This Guide

This section lists the documentation updates for release 25.1.1xx.

Release 25.1.100- G20523-02, April 2025

Relocated the Preface section for improved document structure and readability.

Release 25.1.100- G20523-01, April 2025

- Feature Updates:
  - New Features:
    - \* DNS SRV Based Selection of NRF in NSSF:
      - \* Added the "DNS SRV Based Selection of NRF in NSSF" section to describe the feature.
      - \* Added the following metric in the <u>NSSF Metrics</u> section:
        - \* nrfclient\_nrf\_operative\_status
        - \* nrfclient\_dns\_lookup\_request\_total
      - \* Added the "NrfUri" dimension for nrfclient\_nrf\_operative\_status metric in the <u>Dimensions</u> section:
      - \* Added the following alerts in the <u>NSSF Alerts</u> section:
        - \* OcnssfNrfInstancesInDownStateMajor
        - \* OcnssfAllNrfInstancesInDownStateCritical
    - \* Deleting All Slices in a TAI Using PATCH Remove Operation:
      - \* Added the "Deleting All Slices in a TAI Using PATCH Remove Operation" section to describe the feature.
      - \* Added enhancedPatchBehaviour parameter in CNC Console configuration of NSSF System Option.

#### - Enhancements:

- \* LCI and OCI Headers
  - Updated the feature background and overview details.
  - \* Improved content structure by rearranging existing sections.
  - \* Added details of current scope of this feature and planned enhancements.
- \* <u>Deleting Subscription on 404 SUBSCRIPITON\_NOT\_FOUND Response from AMF</u>
  - \* Updated the feature background and overview details.
  - \* Added advantages and use case flow details.
- General Updates:
  - Updated release number to 25.1.100 throughout the document.
  - Updated the procedure for configuring <u>AMF Resolution</u> using CNC Console to include important cautions that must not be ignored.



- Updated the name of the
  - "ocnssf\_nssaiavailability\_notification\_delete\_on\_subscription\_not\_found" metric by appending "\_total" at the end. The metric name is now updated to "ocnssf\_nssaiavailability\_notification\_delete\_on\_subscription\_not\_found\_to tal" in the NSSF Error Metrics section.
- Added the "Subscription\_removed" dimension for ocnssf\_nssaiavailability\_notification\_delete\_on\_subscription\_not\_found\_tot al metric in <u>Dimensions</u> section.

# Introduction

### 1.1 Overview

This section describes the role of Oracle Communications Network Slice Selection Function (NSSF) in the 5G Service Based Architecture (SBA).

Network slices enable the users to select customized networks with different functionalities (such as mobility) and performance requirements (such as latency, availability and reliability). Network slices differ in features supported and network function optimizations. In such cases, network slices may have different S-NSSAIs with different slice and service types. The user can deploy instances of multiple network slices delivering the same features but for different groups of User Equipments (UEs). These instances deliver different committed services as they are dedicated to a customer, the network slices may have different S-NSSAIs with the same slice or service type but different slice differentiators. The NSSF fulfills the requirement for determining the individual network function pertaining to a slice.

#### (i) Note

The performance and capacity of the NSSF system may vary based on the call model, Feature or Interface configuration, and underlying CNE and hardware environment.

NSSF is a functional element that supports the following functionalities:

- NSSF enables the Access and Mobility Management Function (AMF) to perform initial registration and Protocol Data Unit (PDU) session establishment.
- AMF can retrieve NRF, NSI ID, and target AMFs as part of UE initial registration and PDU establishment procedure.
- NSSF uses an NF Service Consumer (AMF) to update the S-NSSAI(s) that AMF supports and notifies of any changes in the status.
- NSSF selects the network slicing instance (NSI) and determines the authorized Network Slice Selection Assistance Information (NSSAIs) and AMF to serve the UE.
- NSSF interaction with NRF allows retrieving specific NF services to be used for registration request.

NSSF provides the following information when queried by the AMF:

- Allowed NSSAIs
- Configured NSSAIs
- Restricted NSSAIs
- Candidate AMF List (in case of registration)
- Network Slice instance ID (for PDU session establishment)
- Slice-level NRF information (for PDU Connectivity)



NSSF supports the above functions through the following NSSF services:

- NSSelection service (Nnssf\_NSSelection): This service is used by an NF Service
  Consumer (AMF) to retrieve the information related to network slice. It enables network
  slice selection in the serving Home Public Land Mobile Network (HPLMN).
- NSAvailability Service (Nnssf\_NSAvailability): This service stores and maintains list of supported S-NSSAIs per TA. It allows NF service Consumer (AMF) to update and subscribe the above data and get notifications for any addition or deletion of supported S-NSSAIs.

#### **NSSF Availability**

Oracle Communications Cloud Native Core, Network Slice Selection Function (NSSF) availability is dependent on many factors. NSSF applications are designed to achieve 99.999% availability, according to the applicable Telecommunications Industry Association TL9000 standards, with the following deployment requirements:

- Deploy on a Cloud Native Environment with at least 99.999% Availability.
- Deploy with n + k application redundancy, where k is greater than or equal to one.
- Maintain production software within n-3 software releases, where n is the current general availability release.
- Apply bug fixes, critical patches, and configuration recommendations provided by Oracle promptly.
- Maintain fault recovery procedures external to the applications for the reconstruction of lost or altered files, data, programs, or Cloud Native environment.
- Install, configure, operate, and maintain NSSF as per Oracle's applicable installation, operation, administration, and maintenance specifications.
- Maintain an active support contract and provide access to the deployed NSSF and your personnel to assist Oracle in addressing any outage.

NSSF availability is measured for each calendar year and is calculated as follows:

Table 1-1 Measuring NSSF Availability

Availability	Description
Planned Product Availability	(Product available time in each month) less (Excluded Time (defined below) in each month).
Actual Product Availability	(Planned Product Availability) less (any Unscheduled Outage).
Product Availability Level	(Actual Product Availability across all Production instances divided by Planned Product Availability across all Production instances) x 100.



#### (i) Note

#### **Excluded Time means:**

- Scheduled maintenance time.
- Lack of power or backhaul connectivity, except to the extent that such lack of backhaul connectivity was caused directly by the CNC NF.
- Hardware failure.
- Issues arising out of configuration errors or omissions.
- Failures caused by third-party equipment or software not provided by Oracle.
- Occurrence of any event under Force Majeure.
- Any time associated with failure to maintain the recommended architecture and redundancy model requirements above.

## 1.2 References

- Oracle Communications Cloud Native Core, Network Slice Selection Function Installation, Upgrade, and Fault Recovery Guide
- Oracle Communications Cloud Native Core, Network Slice Selection Function Network Impact Report
- Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide
- Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide
- Oracle Communications Cloud Native Core, Automated Testing Suite Guide
- Oracle Communications Cloud Native Core, cnDBTier User Guide
- Oracle Communications Cloud Native Core, Data Collector User Guide

# **NSSF Supported Services**

This chapter includes information about the services supported by NSSF.



#### (i) Note

The performance and capacity of the NSSF system may vary based on the call model, Feature or Interface configuration, and underlying CNE and hardware environment.

# 2.1 Network Slice Selection Service

The Network Slice Selection service is identified by the service operation name, Nnssf\_NSSelection. This service supports the GET request during the following procedures by UE:

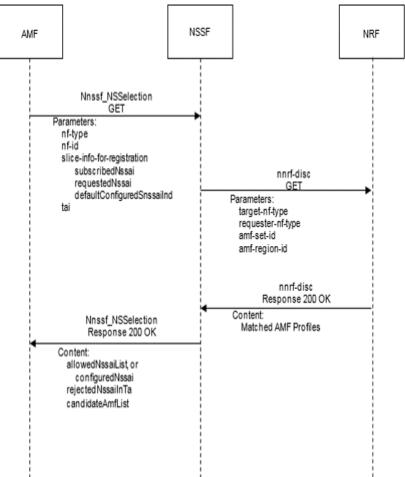
#### **Initial Registration:**

When NSSF is able to find authorized network slice information for the requested network slice, the response includes a payload containing at least the Allowed NSSAI, target AMF Set, or the list of candidate AMF(s).

Following diagram illustrates the procedure of initial registration:



Figure 2-1 Initial Registration



The AMF sends a GET request to the NSSF.

The AMF GET request must include:

- Subscribed S-NSSAIs (with an indication if marked as default S-NSSAI)
- Any Allowed NSSAI

The query parameters may also contain:

- Requested NSSAI
- Mapping of requested NSSAI to configured NSSAI for the HPLMN
- Mapping to the Configured NSSAI for the HPLMN
- PLMN ID of the Subscription Permanent Identifier (SUPI)
- UE's current Tracking Area
- NF type of the NF service consumer
- AMF ID
- Based on the query parameters mentioned above, local configuration, and locally available information, including Radio Access Network (RAN) capabilities obtained by the current Tracking Area for the UE, NSSF does the following:



- It selects the Network Slice instance(s) to serve the UE. When multiple Network Slice instances in the UE's Tracking Areas are able to serve a given S-NSSAI (based on operator's configuration), NSSF selects one slice to serve the UE, or defer the selection of the Network Slice instance until a NF or service within the Network Slice instance needs to be selected.
- It determines the target AMF set to be used to serve the UE or based on configuration, the list of candidate AMF(s), possibly after querying the NRF.
- The AuthorizedNetworkSliceInfo response for UE-registration must mandatorily include both AllowedNSSAI and Candidate AMF list or target amfset. The AllowedNSSAI is computed by taking into account the input request and applying operator policies as specified in 3GPP Spec 29.531 Release 15.5.
- NSSF calculates ConfiguredNSSAI by determining the intersection of S-NSSAI(s) in ConfiguredNSSAI for the PLMN (operator configured) and S-NSSAI(s) in SubscribedNSSAI (from the message indicating SubscribedNSSAI by the UE).
- It determines the Allowed NSSAI(s) for the applicable Access Type(s), taking also into account the availability of the Network Slice instances that are able to serve the S-NSSAI(s) in the Allowed NSSAI and the current UE's tracking areas.
- Based on operator configuration, the NSSF determines the NRF(s) to be used to select NFs or services within the selected Network Slice instance(s).
- When the NSSF locates the authorized network slice information for the requested network, NSSF sends Discovery Request for AMF to NRF.
- The NRF responds with the list of all candidate AMFs to NSSF.
- The NSSF returns to the current AMF the Allowed NSSAI for the applicable Access Type(s), the target AMF Set, or the list of candidate AMF(s) based on configuration.
  - NSSF returns the NRF(s) to be used to select NFs/services within the selected Network Slice instance(s) and the NRF to be used to determine the list of candidate AMF(s) from the AMF Set.
  - NSSF returns NSI ID(s) to be associated to the Network Slice instance(s) corresponding to certain S-NSSAIs.
  - NSSF also returns the rejected S-NSSAI(s) and the Configured NSSAI for the Serving PLMN.
- Candidate AMF selection by NSSF:
  - In response to the Initial Registration request, NSSF responds with AMFs that support the Authorized NSSAI in the specified TAI. NSSF prioritizes runtime data, and if a suitable match is not found, it falls back to operator-configured data. Runtime data consists of NsAvailabilityData sent by the AMF.
  - This approach is chosen to ensure that NSSF responds with consideration of dynamic data. In scenarios where sufficient data is not present (for example, AMFs have not sent an Availability Update), NSSF relies on operator-configured data.
  - This ensures that a response is provided when NSSF is just initialized and does not have Availability Data. The recommendation is that AMFs should have a configuration to update NsAvailability information with NSSF, so that NSSF can respond based on dynamic data.

#### **PDU Session Establishment:**

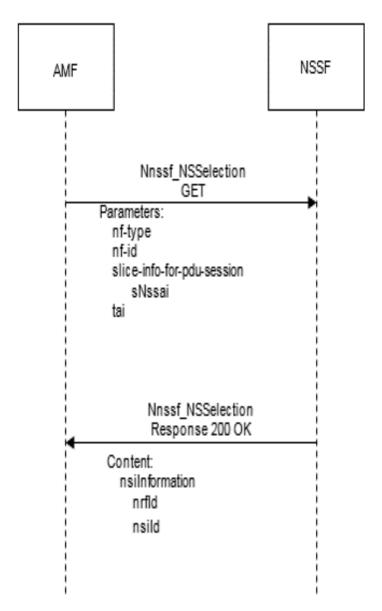
When the NSSF receives a PDU-Session establishment request from the NF consumer, it determines the network slice that can serve the requested S-NSSAI based on the user



configured policies, and responds with the URL of the NRF that manages the Slice and Slice ID of the matching Network slice computed.

The PDU session establishment in a Network Slice to a Data Network (DN) allows data transmission in a Network Slice. A PDU Session is associated with a S-NSSAI and a Data Network Name (DNN). Following diagram illustrates the procedure of PDU Session Establishment:

Figure 2-2 PDU Session Establishment



The following is performed for PDU Session Establishment:

• If the AMF is not able to determine the appropriate NRF to query for the S-NSSAI provided by the UE, the AMF sends a GET request to the NSSF. The AMF queries the NSSF with this specific S-NSSAI, the NF type of the NF service consumer, Requester ID, PLMN ID of the SUPI, and the location information.



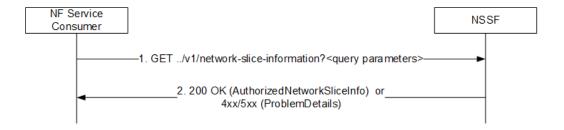
 The NSSF determines and returns the appropriate NRF to be used for selecting NFs or services within the selected Network Slice instance. The NSSF may also return an NSI ID identifying the Network Slice instance to use for this S-NSSAI.
 When a PDU Session for a given S-NSSAI is established using a specific Network Slice instance, the cloud native provides the RAN with S-NSSAI corresponding to this Network Slice instance, which enables the RAN to perform access specific functions.

#### **UE-Config-Update:**

When the UDM updates the Subscribed S-NSSAI(s) to the serving AMF, based on configuration in the AMF, the NSSF determines the mapping of the Configured NSSAI for the serving PLMN and Allowed NSSAI to the Subscribed S-NSSAI(s).

Following diagram illustrates the procedure of UE-Config-Update:

Figure 2-3 UE-Config-Update



The following is performed for UE-Config-Update:

- The AMF sends a UE-Config-Update (GET) request to NSSF. NSSF checks and validates
  the Subscribed S-NSSAI(s), Requested S-NSSAI(s), PLMN ID of the SUPI, TAI, NF type,
  and NF instance ID. If message is valid, NSSF searches for Allowed S-NSSAI list based
  on policy configuration and input parameters.
- NSSF responds with "200 OK with AuthorizedNetworkSliceInfo" if it finds a match.
- The AuthorizedNetworkSliceInfo response for UE-Config-Update must mandatorily include both AllowedNSSAI and ConfiguredNSSAI. The AllowedNSSAI is computed by taking into account the input request and applying operator policies as specified in 3GPP Spec 29.531 Release 15.5.
- NSSF calculates ConfiguredNSSAI by determining the intersection of S-NSSAI(s) in ConfiguredNSSAI for the PLMN (operator configured) and S-NSSAI(s) in SubscribedNSSAI (from the message indicating SubscribedNSSAI by the UE).
- NSSF responds with error code if it finds incorrect parameter validation.

# 2.2 NSSAI Availability Service

The NSSAI Availability service is identified by the service name, Nnssf\_NSSAIAvailability. The following operations are defined for this service:

- Update Service Operation
- Subscribe Service Operation
- Unsubscribe Service Operation

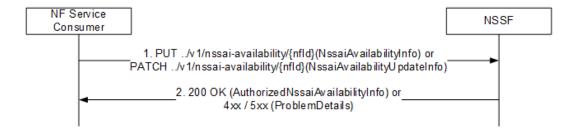


- Notify Service Operation
- Delete Service Operation

#### 1. Update Service Operation

The AMF uses this operation to update the NSSF with the supported S-NSSAI(s) on a per TA basis and to get informed on the S-NSSAIs available per TA (unrestricted) and the restricted S-NSSAI(s) per PLMN in that TA in the serving PLMN of the UE.

Figure 2-4 Update the S-NSSAIs the AMF supports per TA



- The NF service consumer (for example, AMF) sends a PUT request to NSSF with NSSAI availability information, identified by {nfId}, using NssaiAvailabilityInfo.
  - The message contains a list of S-NSSAIs supported by the AMF on a per TA basis.
- NSSF checks (from operator configuration) if the S-NSSAIs are allowed in the TAI and responds with the S-NSSAIs that are authorized by NSSF and supported by the AMF for each TAI.
- If none of the S-NSSAIs in the request are authorized by NSSF (either not configured by the operator or restricted by the operator), NSSF responds with a 403 message.
- In cases where ONSSAI is true and the AMF sends a TaiList or TaiRangeList additionally:
  - NSSF computes the authorized S-NSSAIs for the main TAI.
  - NSSF includes only those TaiList or TaiRangeList that support all the authorized S-NSSAIs from the request in the response.
- NSSF supports HTTP PATCH for NSAvailability Update.
- Upon receiving a PUT or PATCH message, NSSF stores or updates the list in the session database.
- NSSF authorizes the list based on NSSAI authorization rules and responds with the list of allowed S-NSSAIs for that AMF on a per TAI basis as per the request.

#### 2. Subscribe Service Operation

The Subscribe operation is used by NF Service Consumer (AMF) to get the notifications for any change in NSSAI availability information.



Figure 2-5 Subscription creation



- AMF sends a POST request to NSSF with notification URL and a list of TAIs as JSON body.
- NSSF stores the subscription request and responds with the list of allowed S-NSSAI(s)
  per TAI in the request. NSSF also returns a subscription-id and expiry (duration up to
  which NSSF sends notifications for any change in the status of Grant of S-NSSAI for
  subscribed TAI(s)).

#### 3. Unsubscribe Service Operation

The Unsubscribe service operation is used by AMF to unsubscribe to a notification of any previously subscribed changes to the NSSAI availability information.

Figure 2-6 Unsubscribe a Subscription



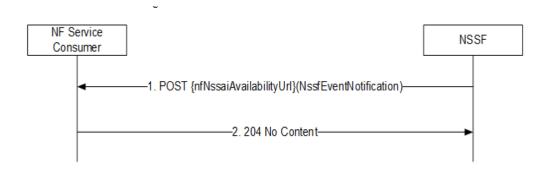
- AMF sends a Delete request to NSSF with subscription-id.
- NSSF checks for active subscription with the id and if found, deletes the subscription and responds with the message 204.

#### 4. Notify Service Operation



The Notify service operation is used by the NSSF to update the AMF with any change in status, on a per TA basis, of the S-NSSAIs available per TA (unrestricted) and the S-NSSAIs restricted per PLMN in that TA in the serving PLMN of the UE.

Figure 2-7 Update the AMF with any S-NSSAI restricted per TA

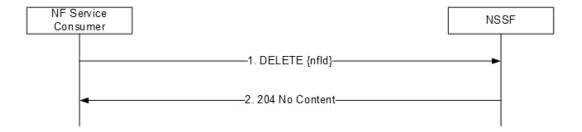


- NSSF sends notification to subscribed AMF when one or more following conditions are true:
  - There is change at Grant rules on S-NSSAI corresponding to one or more of TAIs subscribed by AMF.
  - An S-NSSAI has been added or deleted for one or more of TAIs subscribed by AMF.

#### 5. Delete Service Operation

The AMF uses this operation to delete the NSSAI Availability information stored for that AMF in the NSSF.

Figure 2-8 Delete the NSSAI Availability Information at NSSF

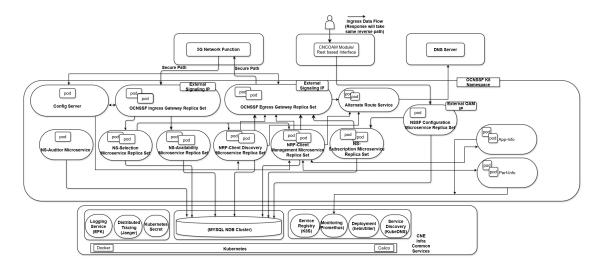


- The NF service consumer (For example: AMF) sends a DELETE request to NSSF with nfId.
- The NSSF searches in session database for the NSAvailability data corresponding to nfId and deletes them.

# **NSSF Architecture**

NSSF comprises of various microservices deployed in Kubernetes based Cloud Native Environment (CNE), for example: Oracle Communications Cloud Native Core, Cloud Native Environment (CNE). CNE provides some common services like logs, metrics data collection, analysis, graphs, and charts visualization, etc. The microservices integrate with these common services and provide them with the necessary data.

The following diagram describes the overall architecture of the NSSF:



The architecture has the following components:

#### 1. NSSelection Service

The Network Slice Selection service is identified by the service operation name, Nnssf\_NSSelection. This service supports the GET request during the following procedures by UE:

#### a. Initial Registration:

When the NSSF is able to find authorized network slice information for the requested network slice, the response body includes a payload body containing at least the Allowed NSSAI, target AMF Set, or the list of candidate AMF(s).

- The AMF sends a GET request to the NSSF. The AMF GET request must include:
  - Subscribed S-NSSAIs (with an indication if marked as default S-NSSAI)
  - Any Allowed NSSAI
- The guery parameters may also contain:
  - Requested NSSAI
  - Mapping of requested NSSAI to configured NSSAI for the HPLMN
  - Mapping to the Configured NSSAI for the HPLMN
  - PLMN ID of the Subscription Permanent Identifier (SUPI)
  - UE's current Tracking Area



- NF type of the NF service consumer
- AMF ID
- Based on the query parameters mentioned above, local configuration, and other locally available information including Radio Access Network (RAN) capabilities made available by the current Tracking Area for the UE, the NSSF does the following:
  - It selects the Network Slice instance(s) to serve the UE. When multiple
    Network Slice instances in the UE's Tracking Areas are able to serve a given
    S-NSSAI, based on operator's configuration, the NSSF may select one of
    them to serve the UE, or the NSSF may defer the selection of the Network
    Slice instance until a NF or service within the Network Slice instance needs to
    be selected.
  - It determines the target AMF set to be used to serve the UE or based on configuration, the list of candidate AMF(s), possibly after querying the NRF.
  - The AuthorizedNetworkSliceInfo response for UE-registration must mandatory include both AllowedNSSAI and Candidate AMF list or target amfset. The AllowedNSSAI is computed by taking into account the input request and applying operator policies as specified in 3GPP Spec 29.531 Release 15.5.
  - NSSF calculates ConfiguredNSSAI by determining the intersection of S-NSSAI(s) in ConfiguredNSSAI for the PLMN (operator configured) and S-NSSAI(s) in SubscribedNSSAI (from the message indicating SubscribedNSSAI by the UE).
  - It determines the Allowed NSSAI(s) for the applicable Access Type(s), taking also into account the availability of the Network Slice instances that are able to serve the S-NSSAI(s) in the Allowed NSSAI in the current UE's tracking areas.
  - Based on operator configuration, the NSSF may determine the NRF(s) to be used to select NFs or services within the selected Network Slice instance(s).
- When the NSSF is able to find authorized network slice information for the requested network, NSSF sends Discovery Request for AMF to NRF.
- The NRF responds with list of candidate AMFs to NSSF.
- The NSSF returns to the current AMF the Allowed NSSAI for the applicable Access Type(s), the target AMF Set, or, based on configuration, the list of candidate AMF(s).
  - NSSF returns the NRF(s) to be used to select NFs/services within the selected Network Slice instance(s) and the NRF to be used to determine the list of candidate AMF(s) from the AMF Set.
  - NSSF returns NSI ID(s) to be associated to the Network Slice instance(s) corresponding to certain S-NSSAIs.
  - NSSF also returns the rejected S-NSSAI(s) and the Configured NSSAI for the Serving PLMN.

#### b. PDU Session Establishment:

When the NSSF receives PDU-Session establishment request from the NF consumer, NSSF determines the network slice which can serve the requested S-NSSAI, based on the user configured policies, and responds with the URL of NRF which manages to the Slice and/or Slice ID of the matching Network slice computed.

The PDU session establishment in a Network Slice to a Data Network (DN) allows data transmission in a Network Slice. A PDU Session is associated with a S-NSSAI and a Data Network Name (DNN).



The following is performed for PDU Session Establishment:

- If the AMF is not able to determine the appropriate NRF to query for the S-NSSAI provided by the UE, the AMF sends a GET request to the NSSF. The AMF queries the NSSF with this specific S-NSSAI, the NF type of the NF service consumer, Requester ID, PLMN ID of the SUPI, and the location information.
- The NSSF determines and returns the appropriate NRF to be used to select NFs or services within the selected Network Slice instance. The NSSF may also return an NSI ID identifying the Network Slice instance to use for this S-NSSAI. When a PDU Session for a given S-NSSAI is established using a specific Network Slice instance, the cloud native provides to the RAN the S-NSSAI corresponding to this Network Slice instance to enable the RAN to perform access specific functions.

#### c. UE-Config-Update:

When the UDM updates the Subscribed S-NSSAI(s) to the serving AMF, based on configuration in this AMF, the NSSF determines the mapping of the Configured NSSAI for the serving PLMN and Allowed NSSAI to the Subscribed S-NSSAI(s).

The following is performed for UE-Config-Update:

- The AMF sends a UE-Config-Update (GET) request to NSSF. NSSF checks and validates the Subscribed S-NSSAI(s), Requested S-NSSAI(s), PLMN ID of the SUPI, TAI, NF type, and NF instance ID. If message is valid, NSSF searches for Allowed S-NSSAI list based on policy configuration and input parameters.
- NSSF responds with 200 OK with AuthorizedNetworkSliceInfo in case NSSF finds a match.
- The AuthorizedNetworkSliceInfo response for UE-Config-Update must mandatorily include both AllowedNSSAI and ConfiguredNSSAI. The AllowedNSSAI is computed by taking into account the input request and applying operator policies as specified in 3GPP Spec 29.531 Release 15.5.
- NSSF calculates ConfiguredNSSAI by determining the intersection of S-NSSAI(s) in ConfiguredNSSAI for the PLMN (operator configured) and S-NSSAI(s) in SubscribedNSSAI (from the message indicating SubscribedNSSAI by the UE).
- NSSF responds with error code in case of incorrect parameter validation.

#### 2. NS Availability Service

This microservice supports NSAvailability service of NSSF as per 29.531. This microservice stores subscriptions and AMF data.

The NSSAI Availability service is identified by the service name, Nnssf\_NSSAIAvailability. For the Nnssf\_NSSAIAvailability service the following service operations are defined:

- Update Service Operation
- Subscribe Service Operation
- Unsubscribe Service Operation
- Delete Service Operation

#### 3. NS Subscription Service

This micro-service sends notifications based on Subscribed Events through NSAvailability.

Notifications are sent to Subscribed AMFs to signify changes in Authorization state with respect to S-NSSAIs on TAI as per 3GPP TS 29.531

The Notify operation is used by the NSSF to update the AMF with any change in status, on a per TA basis, of the S-NSSAIs available per TA (unrestricted) and the S-NSSAIs restricted per PLMN in that TA in the serving PLMN of the UE.



- NSSF sends notification to subscribed AMF when one or more following conditions are true:
  - There is change at Grant rules on S-NSSAI corresponding to one or more of TAIs subscribed by AMF.
  - An S-NSSAI has been added or deleted for one or more of TAIs subscribed by AMF.

#### 4. NS Auditor Service

This microservice is a timed auditor, which removes stale records from NSSF.

#### What is a stale record?

In georedundant scenarios, tables in State Database (stateDB) maintain a column siteId, which identifies owner site of that record. There could be georedundancy scenarios when similar records can be owned by two sites, where the older record is termed as the stale record.

NS Auditor is used in georedundancy scenarios where subscription is owned by one site, but the Patch is received on other site. This leads to creation of two records for same subscription owned by each site. Ns-Auditor detects this and removes the old stale record to ensure subscription is owned at a single site only.

#### For example:

- Site-1 receives Subscription POST.
- Site-1 creates a record, rec-1, for subscription with owner as site-1.
- If AMF gets disconnected with the site-1, site-1 goes down, or SCP makes a routing decision based on congestion, the subscription PATCH is received on site-2.
- Site-2 creates a new record, rec-2, for subscription with new owner as site-2.
- Now, rec-2 for site-2 is same as rec-1 for site1.
- NS Auditor detects this and deletes rec-1.
- Site-2 becomes the owner of the subscription, and receives the latest patch.

#### 5. NS Configuration Service

This microservice is responsible for configuring policy rules. It implements a REST messaging server that receives configuration HTTP messages, validates and stores the configuration in the database.

#### 6. NRF Client Manegement

This microservice registers with the NRF and sends periodic heartbeats, also maintains subscriptions with NRF for AMF sets.

- NRF Registration and Heartbeat: Once NSSF is registered with NRF, NSSF contacts
  the NRF periodically. First the registration profile is configured using helm. Then the
  performance service calculates load and capacity of NF. NS registration requests the
  load and capacity from performance service and sends it to NRF with heartbeat.
- NRF Subscription: NSSF subscribes to NRF for AMF based on the Target AMF Set and Region ID for registration and deregistration and load update.

#### 7. NRF Client Discovery Microservice

This microservice plays a crucial role in managing discovery requests within a network. Its primary function is to handle on-demand service discovery and efficiently manage interactions with the Network Repository Function (NRF).

 Discovery: Performs service discoveries as required. When a service or function needs to be identified or connected, the discovery component initiates the process.



- Handling Requests: Handles requests directed towards the NRF, which is responsible for registering and providing information on network functions and their services.
- Response Processing: Once a response is received from the NRF, the microservice processes this information and prepares it for the requesting service.
- 8. App-Info: This microservice monitors application (microservice) health and status.
- 9. Perf-Info: This microservice monitors application (microservice) capacity and load status.
- Configuration Server: This service performs the database abstraction for storage and retrieval of NSSF configuration.
- 11. Alternate Route Service: Alternate Route Service (ARS) is a microservice designed to efficiently find and provide alternate network routes. It employs a tiered approach, beginning with a fast cache lookup. If the desired route isn't cached or the cached entry is outdated, ARS checks predefined static mappings. It then queries DNS-SRV (if configured), updates its cache, and returns a response indicating success or failure. On success, ARS provides a list of alternate FQDNs (Fully Qualified Domain Names). These alternate routes can be configured either statically (using Helm charts) or dynamically (through DNS-SRV). Essentially, ARS prioritizes speed and efficiency in finding alternate routes by leveraging caching and fallback mechanisms.

#### 12. Ingress Gateway Service

This microservice is an entry point for accessing NSSF supported service operations and provides the functionality of an OAuth validator.

#### 13. Egress Gateway Service

This microservice is responsible to route NSSF initiated egress messages to other NFs.

#### Note

For more information on Ingress and Egress Gateway, see *Oracle Communications Cloud Native Core*, *Cloud Native Environment User Guide*.

# **NSSF Supported Features**

This section explains about the NSSF supported features.



#### (i) Note

The performance and capacity of the NSSF system may vary based on the call model, Feature or Interface configuration, and underlying CNE and hardware environment.

## 4.1 DNS SRV Based Selection of NRF in NSSF

Currently, the NSSF selects the NRF for communication based on static configurations set by the operator. However, there are scenarios where the NRF may go down for various reasons. In such cases, the NSSF, which relies on the NRF for specific communications, may experience service disruptions.

The DNS SRV based selection of NRF in NSSF feature enhances network resilience by utilizing the NRF's georedundancy capabilities to handle potential site failures. This setup enables Network Functions (including NSSF) to continue operating smoothly by redirecting traffic from a primary NRF to a secondary NRF when a failure occurs at the primary site.

This feature allows the NSSF to dynamically select NRF instances based on real-time availability and site redundancy through DNS SRV configurations. In addition to the static configurations by operators, the NSSF can now resolve NRFs using DNS SRV-based Fully Qualified Domain Names (FQDNs). The NSSF is configured with a primary NRF and multiple fallback NRFs, which take over if the primary NRF becomes unreachable.

The nrf-client uses the Alternate Route Service, which helps the Network Slice Selection Function (NSSF) find and select different Network Repository Functions (NRFs) by using DNS SRV-based lookups. This service allows the NSSF to translate Fully Qualified Domain Names (FQDNs) or virtual FQDNs into alternate NRF addresses. This setup enables the NSSF to prioritize and adjust connections to different NRFs based on specific service needs.

**How Alternate Routing Works** 



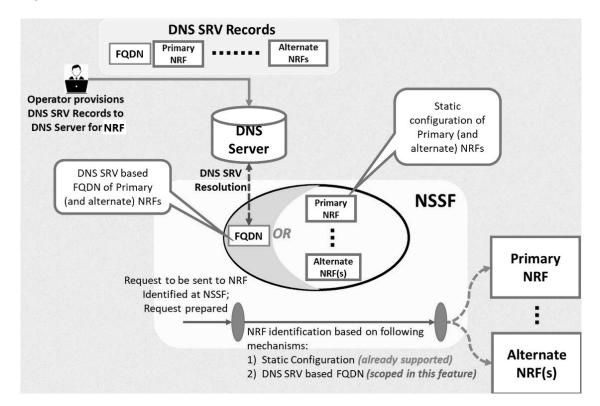


Figure 4-1 Call Flow of DNS SRV Based Selection of NRF in NSSF

- **1. DNS SRV Record Lookup**: The NSSF starts by using DNS SRV records to fetch information about the FQDN of the primary NRF and alternate NRFs.
- DNS Query: The DNS server receives a query from the NSSF to resolve the FQDN of the primary NRF or alternate NRFs.
- **3. DNS Resolution**: The DNS server resolves the FQDN and returns the resolved address for the primary NRF or one of the alternate NRFs.
- 4. NRF Identification by NSSF: Based on the FQDN provided by DNS, the NSSF identifies the primary NRF to communicate with. If the primary NRF is unavailable, the NSSF uses one of the alternate NRFs.
- 5. **Static Configuration Option**: Alternatively, the NSSF can rely on a static configuration of the primary and alternate NRFs instead of querying DNS.
- **6. Request Preparation**: Once an NRF (primary or alternate) is identified, the NSSF prepares the request to be sent to the selected NRF.
- Communication with NRF: The NSSF sends the request to the identified NRF, enabling further processing or service access.

The Alternate Route Service utilizes DNS SRV records to look up virtual FQDNs. The Egress Gateway queries this service to retrieve a list of alternate NRFs along with their priorities. Based on these priorities, the Egress Gateway reroutes requests to other NRF instances as required.

#### **Key Operations of nrf-client for High Availability**

The NSSF uses a component called nrf-client to facilitate communication with NRFs. The nrf-client performs critical functions to ensure NRFs remain available and responsive:



- Traffic Routing: The nrf-client directs all requests (for example, registration, updates, heartbeats, and discovery) to the primary NRF.
- 2. Failure Detection: If the primary NRF cannot be reached, the nrf-client detects the failure either through routing errors or by receiving a "503 Service Unavailable" response, which may include a "Retry-After" interval.
- 3. Failover and Retry: If the primary NRF fails, the nrf-client temporarily marks it as unavailable and reroutes traffic to a secondary NRF based on DNS SRV priority. The retry interval, specified in the "Retry-After" header or a preset interval, determines when the nrf-client reattempts contact with the primary NRF.
  - During this interval, all requests are routed to the secondary NRF.
  - Once the retry interval expires, the nrf-client resumes attempts to communicate with the primary NRF.

#### (i) Note

The nrf-client only tries to connect with each NRF once. If it cannot send a request to an NRF, it returns a "503 Service Unavailable" response to the NSSF. This one-time attempt helps avoid repeated failures and ensures alternate routing to secondary NRFs when needed.

#### Managing DNS SRV Based Selection of NRF in NSSF

This section provides information about Helm, REST API, and Cloud Native Configuration Console (CNC Console) configurations required to configure this feature.

#### **Enable:**

You can enable this feature using Helm configurations. To enable this feature, it is mandatory to configure the following Helm parameters as given below:

enableVirtualNrfResolution=true
virtualNrfFqdn=nrfstub.changeme-ocats.svc
virtualNrfScheme=http

#### (i) Note

If enableVirtualNrfResolution is set to false, the nrf-client uses primaryNrfApiRoot and secondaryNrfApiRoot configurations to register with the NRF. However, if enableVirtualNrfResolution is set to true and an incorrect virtualNrfFqdn is configured, the nrf-client does not fallback to primaryNrfApiRoot and secondaryNrfApiRoot for registration, resulting in a registration failure.

Apart from this, for more information on other Helm parameters supported for this feature, see Helm section below.

#### Configure

#### Helm



To enable this feature, configure the following nrf-client Helm parameters in ocnssf\_custom\_values\_25.1.100.yaml file.

```
nrf-client:
  # This config map is for providing inputs to NRF-Client
  configmapApplicationConfig:
    &configRef
    profile: |-
      [appcfq]
      primaryNrfApiRoot=nrf-stubserver.changeme-ocats:8080
      secondaryNrfApiRoot=nrf-stubserver.changeme-ocats:8080
      nrfScheme=http
      retryAfterTime=PT120S
      nrfClientType=NSSF
      nrfClientSubscribeTypes=
      appProfiles=[{
  "nfInstanceId": "9faf1bbc-6e4a-4454-a507-aef01a101a01",
  "nfType": "NSSF",
  "nfStatus": "REGISTERED",
  "heartBeatTimer": 30,
  "fqdn": "ocnssf-nsgateway.ocnssf.svc",
  "priority": 1,
  "capacity": 1,
  "load": 2,
  "plmnList": [
      "mcc": "311",
      "mnc": "480"
  ],
  "nfSetIdList": [
    "setEast.nssfset.5qc.mnc480.mcc311"
  "locality": "rcnltxekloc1",
  "nfServices": [
      "serviceInstanceId": "92d59bfc-e5d6-47f5-a26b-3a03facdebcc",
      "serviceName": "nnssf-nsselection",
      "versions": [
          "expiry": null,
          "apiFullVersion": "1.0.0",
          "apiVersionInUri": "v1"
      ],
      "scheme": "http",
      "nfServiceStatus": "REGISTERED",
      "fqdn": "ocnssfl-ingress-gateway.ocnssf.svc",
      "interPlmnFqdn": null,
      "ipEndPoints": [
          "ipv4Address": "10.224.45.178",
          "transport": "TCP",
          "port": 80
```



```
"allowedNfTypes": [
        "AMF",
        "NSSF"
      "priority": 1,
      "capacity": 1,
      "load": 2
      "serviceInstanceId": "d33728cd-6e21-434b-bc5a-ed69bc612377",
      "serviceName": "nnssf-nssaiavailability",
      "versions": [
          "expiry": null,
          "apiFullVersion": "1.0.0",
          "apiVersionInUri": "v1"
      ],
      "scheme": "http",
      "nfServiceStatus": "REGISTERED",
      "fqdn": "ocnssf2-ingress-gateway.ocnssf.svc",
      "interPlmnFqdn": null,
      "ipEndPoints": [
          "ipv4Address": "10.224.45.179",
          "transport": "TCP",
          "port": 80
      ],
      "allowedNfTypes": [
        "AMF",
        "NSSF"
      "priority": 1,
      "capacity": 1,
      "load": 2
 ]
}]
      enableF3=true
      enableF5=true
      renewalTimeBeforeExpiry=3600
      validityTime=30
      enableSubscriptionAutoRenewal=true
      nfHeartbeatRate=80
      acceptAdditionalAttributes=false
      retryForCongestion=5
      enableVirtualNrfResolution=true
      virtualNrfFqdn=nrfstub.changeme-ocats.svc
      virtualNrfScheme=http
```





This is a sample configuration. Modify the parameters as per your setup. For detailed information about the Helm parameters, see the "Customizing NSSF" section in the Oracle Communications Cloud Native Core, Network Slice Selection Function Installation, Upgrade, and Fault Recovery Guide.

#### **REST API**

There are no REST API configurations required for this feature.

#### **CNC Console**

There is no option to enable or disable this feature using CNC Console.

#### **Observe**

#### **Metrics**

The following metric is available for this feature:

- nrfclient\_nrf\_operative\_status
- nrfclient\_dns\_lookup\_request\_total

For more information about metrics, see **NSSF Metrics** section.

#### **KPIs**

There are no new KPIs for this feature.

#### **Alerts**

The following alerts are available for this feature:

- OcnssfNrfInstancesInDownStateMajor
- OcnssfAllNrfInstancesInDownStateCritical

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- Collect the logs: For more information on how to collect logs, see Oracle Communications
  Cloud Native Core, Network Slice Selection Function Troubleshooting Guide.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

# 4.2 Deleting All Slices in a TAI Using PATCH Remove Operation

In the context of 5G networks, network slicing is a crucial feature. It allows the network to be divided into multiple virtual slices, each optimized for a specific type of service or use case. These slices are defined by S-NSSAIs (Single Network Slice Selection Assistance Information) and are associated with specific geographical areas, referred to as Tracking Areas (TAIs).

The Access and Mobility Function (AMF) is responsible for managing these slices and ensuring that these devices connect to the appropriate slices. The Network Slice Selection Function (NSSF) assists the AMF in managing and selecting these slices.



Currently, when the AMF needs to update or delete slices for specific areas, there are limitations in how the NSSF responds to such requests. This new feature addresses those limitations, improving flexibility and compliance with certain operational requirements.

This feature enhances the NSSF to support specific behaviors for managing network slices through the PATCH operation. This includes the ability to:

- delete all slices for specific areas (TAI, TAIList, TAIRange) using PATCH.
- improve responses for better clarity and compliance, including the use of 204 No Content when all slices in an area are removed.
- allow slices to be added back later through PATCH operations.

#### Old Behavior (Before enabling the feature):

#### Slice Management with PATCH Operations:

- The NSSF responded with 200 OK to all PATCH requests, even if slices were only partially deleted.
- 204 No Content responses were not supported for PATCH operations.
- If no authorized slices were found in a PATCH request, the NSSF returned a 403
   Forbidden response.

#### Deletion Constraints:

- The AMF could only delete all slices across all areas using a DELETE request, which was not always practical.
- The NSSF required at least one slice to remain in a TAI/TAIList/TAIRange, in accordance with existing specifications.

#### **Enhanced Behavior (When the feature is enabled):**

#### Enhanced Deletion Using PATCH:

- The AMF can now issue a PATCH request to delete all slices for a specific TAI/TAIList/ TAIRange.
  - \* If all slices are successfully deleted for the specified area(s), the NSSF responds with 204 No Content.
  - \* If some slices remain after the deletion, the NSSF responds with 200 OK, including the list of remaining slices.

#### Support for Full Deletion Scenarios:

- If a NssaiAvailability PATCH request results in deletion of all slices within a specific TAI:
  - The response can be 200 OK (if other TAIs still have SNSSAIs that are supported by the AMF and authorized by NSSF), or
  - \* 204 No Content (if none of the SNSSAIs supported by the AMF are authorized by NSSF).
- If all slices are deleted across all TAIs, the NSSF responds with 204 No Content.
- Adding Back Slices: After a full deletion, the AMF can re-add slices for the same area using a PATCH operation.

#### Oracle NSSF Specific Handling:

 When processing a PATCH request that removes all slices, the NSSF verifies compliance with the 3GPP specification, which requires at least one slice in an area.



- To support the new feature, the NSSF uses a flag:
  - \* If the flag is enabled, the NSSF can handle and store areas with no slices (for example, storing a null value or a placeholder).
  - \* This ensures the system behaves as expected while complying with 3GPP standards and supporting this feature.

This feature provides flexibility for the AMF to manage slices with greater granularity, targeting specific areas without affecting others, thereby improving efficiency and control. It aligns with 3GPP standard operations for PATCH while introducing flexibility for specific use cases (e.g., FOA). Additionally, returning 204 No Content for full deletions offers a clearer indication that all slices were successfully removed, enhancing system transparency.

#### Managing Deletion of All Slices in a TAI Using PATCH Remove Operation

This section provides information about Helm, REST API, and Cloud Native Configuration Console (CNC Console) configurations required to configure this feature.

#### **Enable:**

You can enable this feature using REST API or CNC Console.

#### Configure

#### Helm

There are no Helm configurations required for this feature.

#### **REST API**

A new boolean parameter, enhancedPatchBehaviour, is added to enable or disable this feature using NSSF System Option API. Update this parameter value as true or false to enable or disable this feature, respectively. By default, it is set to false.

For more information about the REST API configuration, see "NSSF System Options" section in "NSSF REST Specifications" chapter of *Oracle Communications Cloud Native Core*, *Network Slice Selection Function REST Specification Guide*.

#### **CNC Console**

The value of enhancedPatchBehaviour parameter can also be updated using CNC Console interface for NSSF System Option.

#### Observe

#### Metrics

There are no new Metrics for this feature. However, this feature uses the following metric to count the success response messages sent by NSSF for requests for the Nnssf\_NSSAIAvailability service.

ocnssf\_nssaiavailability\_success\_tx\_total

For more information about metrics, see NSSF Metrics section.

#### **KPIs**

There are no new KPIs for this feature.

#### **Alerts**

There are no new alerts for this feature.



#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. Collect the logs: For more information on how to collect logs, see Oracle Communications Cloud Native Core, Network Slice Selection Function Troubleshooting Guide.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

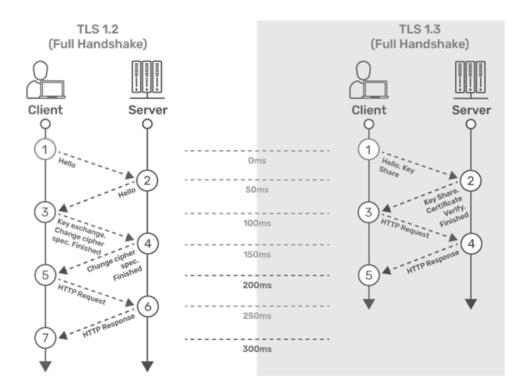
## 4.3 Support for TLS

NSSF uses Hypertext Transfer Protocol Secure (HTTPS) to establish secure connections with Consumer NFs and Producer NFs, respectively. These communication protocols are encrypted using Transport Layer Security (TLS). TLS comprises the following components:

- Handshake Protocol: Exchanges the security parameters of a connection. Handshake messages are supplied to the TLS record layer.
- Record Protocol: Receives the messages to be transmitted, fragments the data into
  multiple blocks, secures the records, and then transmits the result. Received data is
  delivered to higher-level peers.

#### **TLS Handshake**

This section describes the differences between TLSv1.2 and TLSv1.3 and the advantages of TLSv1.3 over TLSv1.2 and earlier versions.



TLSv1.2



- The connection or handshake starts when the client sends a "client hello" message to the server. This message consists of cryptographic information such as supported protocols and supported cipher suites. It also contains a random value or random byte string.
- 2. To respond to the "client hello" message, the server sends the "server hello" message. This message contains the CipherSuite that the server has selected from the options provided by the client. The server also sends its certificate along with the session ID and another random value.
- 3. The client verifies the certificate sent by the server. When the verification is complete, it sends a byte string and encrypts it using the public key of the server certificate.
- 4. When the server receives the secret, both the client and server generate a master key along with session keys (ephemeral keys). These session keys are used to symmetrically encrypt the data.
- 5. The client sends an "HTTP Request" message to the server to enable the server to switch to symmetric encryption using the session keys.
- **6.** To respond to the client's "HTTP Request" message, the server does the same and switches its security state to symmetric encryption. The server concludes the handshake by sending an HTTP response.
- 7. The client-server handshake is completed in two round-trips.

#### TLSv1.3

- The connection or handshake starts when the client sends a "client hello" message to the server. The client sends the list of supported cipher suites. The client also sends its key share for that particular key agreement protocol.
- 2. To respond to the "client hello" message, the server sends the key agreement protocol that it has chosen. The "Server Hello" message comprises the server key share, server certificate, and the "Server Finished" message.
- 3. The client verifies the server certificate, generates keys as it has the key share of the server, and sends the "Client Finished" message along with an HTTP request.
- 4. The server completes the handshake by sending an HTTP response.

The following digital signature algorithms are supported in TLS handshake:

**Table 4-1 Digital Signature Algorithms** 

Algorithm	Key Size (Bits)	Elliptic Curve (EC)
RS256 (RSA)	2048	NA
	4096 This is the recommended value.	NA
ES256 (ECDSA)	NA	SECP384r1 This is the recommended value.

#### Comparison Between TLSv1.2 and TLSv1.3

The following table provides a comparison of TLSv1.2 and TLSv1.3:



Table 4-2 Comparison of TLSv1.2 and TLSv1.3

Feature	TLS v1.2	TLS v1.3
TLS Handshake	<ul> <li>The initial handshake was carried out in clear text.</li> <li>A typical handshake in TLSv1.2 involves the exchange of 5 to 7 packets.</li> </ul>	<ul> <li>The initial handshake is carried out along with the key share.</li> <li>A typical handshake IN TLSv1.3 involves the exchange of up to 3 packets.</li> </ul>
Cipher Suites	Less secure Cipher suites.  Use SHA-256 and SHA-384 hashing  TLS_ECDHE_ECDSA_WITH_AES_25  6_GCM_SHA384  TLS_ECDHE_RSA_WITH_AES_256_ GCM_SHA384  TLS_ECDHE_RSA_WITH_CHACHA20_ POLY1305_SHA256  TLS_ECDHE_ECDSA_WITH_AES_12  8_GCM_SHA256  TLS_ECDHE_RSA_WITH_AES_12  GCM_SHA256	More secure Cipher suites.     Apart from all the ciphers supported for TLSv1.2, the following additional ciphers are supported for only TLSv1.3:     TLS_AES_128_GCM_SH A256     TLS_AES_256_GCM_SH A384     TLS_CHACHA20_POLY13 05_SHA256
Round-Trip Time (RTT)	This has a high RTT during the TLS handshake.	This has low RTT during the TLS handshake.
Perfect Forward Secrecy (PFS)	This doesn't support PFS.	TLSv1.3 supports PFS. PFS ensures that each session key is completely independent of long-term private keys, which are keys that are used for an extended period to decrypt encrypted data.
Privacy	This is less secure, as the ciphers used are weak.	This is more secure, as the ciphers used are strong.
Performance	This has high latency and a less responsive connection.	This has low latency and a more responsive connection.

#### **Advantages of TLSv1.3**

The TLSv1.3 handshake offers the following improvements over earlier versions:

- All handshake messages after the ServerHello are encrypted.
- It improves efficiency in the handshake process by requiring fewer round trips than TLSv1.2. It also uses cryptographic algorithms that are faster.
- It provides better security than TLSv1.2, addressing known vulnerabilities in the handshake process.
- It eliminates data compression.

The following table describes the TLS versions supported on the client and server sides. The last column indicates which version will be used.

#### **TLS Version Used**

When NSSF is acting as a client or a server, it can support different TLS versions.

The following table provides information about which TLS version will be used when various combinations of TLS versions are present between the server and the client.



Table 4-3 TLS Version Used

Client Support	Server Support	TLS Version Used
TLSv1.2, TLSv1.3	TLSv1.2, TLSv1.3	TLSv1.3
TLSv1.3	TLSv1.3	TLSv1.3
TLSv1.3	TLSv1.2, TLSv1.3	TLSv1.3
TLSv1.2, TLSv1.3	TLS v1.3	TLSv1.3
TLSv1.2	TLSv1.2, TLSv1.3	TLSv1.2
TLSv1.2, TLSv1.3	TLSv1.2	TLSv1.2
TLS v1.3	TLSv1.2	Sends an error message. For more information about the error message, see "Troubleshooting TLS Version Compatibilities" section in Oracle Communications Cloud Native Core, Network Slice Selection Function Troubleshooting Guide.
TLSv1.2	TLSv1.3	Sends an error message. For more information about the error message, see "Troubleshooting TLS Version Compatibilities" section in Oracle Communications Cloud Native Core, Network Slice Selection Function Troubleshooting Guide.

#### (i) Note

- If Egress Gateway is deployed with both the versions of TLS that is TLSv1.2 and TLSv1.3, then Egress Gateway as client will send both versions of TLS in the client hello message during the handshake and the server needs to decide which version to be used.
- If Ingress Gateway is deployed with both the version of TLS that is with TLSv1.2 and TLSv1.3, then Ingress Gateway as the server will use the TLS version received from the client in the server hello message during the handshake.
- This feature does not work in ASM deployment.

#### Managing Support for TLSv1.2 and TLSv1.3

#### **Enable:**

This feature can be enabled or disabled at the time of NSSF deployment using the following Helm parameters:

- enableIncomingHttps: This flag is used for enabling/disabling HTTPS/2.0 (secured TLS) in the Ingress Gateway. If the value is set to false, NSSF will not accept any HTTPS/2.0 (secured) traffic. If the value is set to true, NSSF will accept HTTPS/2.0 (secured) traffic. Note: Do not change the &enableIncomingHttpsRef reference variable. For more information on enabling this flag, see the "Enabling HTTPS at Ingress Gateway" section in the Oracle Communications Cloud Native Core, Network Slice Selection Function Installation, Upgrade, and Fault Recovery Guide.
- enableOutgoingHttps: This flag is used for enabling/disabling HTTPS/2.0 (secured TLS) in the Egress Gateway. If the value is set to false, NSSF will not accept any HTTPS/2.0 (secured) traffic. If the value is set to true, NSSF will accept HTTPS/2.0 (secured) traffic.



For more information on enabling this flag, see the "Enabling HTTPS at Egress Gateway" section in the *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Installation*, *Upgrade*, *and Fault Recovery Guide*.

#### Configure

You can configure this feature using Helm parameters.

The following parameters in the Ingress Gateway and Egress Gateway microservices must be customized to support TLSv1.2 or TLSv1.3:

- 1. Generate HTTPS certificates for both the ingress and egress gateways. Ensure that the certificates are correctly configured for secure communication. After generating the certificates, create a Kubernetes secret for each gateway (egress and ingress). Then, configure these secrets to be used by the respective gateways. For more information about HTTPS configuration, generating certificates, and creating secrets, see the "Configuring Secrets for Enabling HTTPS" section in the Oracle Communications Cloud Native Core, Network Slice Selection Function Installation, Upgrade, and Fault Recovery Guide.
- 2. After configuring the secret and applying it to the namespace where NSSF is deployed, perform the following Helm changes for Ingress and Egress gateways in the ocnssf\_custom\_values\_25.1.100.yaml file:
  - Parameters required to support TLSv1.2:
    - service.ssl.tlsVersion indicates the TLS version.
    - cipherSuites indicates supported cipher suites.
    - allowedCipherSuites indicates allowed cipher suites.
  - Parameters required to support TLSv1.3:
    - service.ssl.tlsVersion indicates the TLS version.
    - cipherSuites indicates the supported cipher suites.
    - allowedCipherSuites indicates the allowed cipher suites.
    - clientDisabledExtension is used to disable the extension sent by messages originating from clients during the TLS handshake with the server.
    - serverDisabledExtension is used to disable the extension sent by messages originating from servers during the TLS handshake with the client.
    - tlsNamedGroups is used to provide a list of values sent in the supported\_groups extension. These are comma-separated values.
    - clientSignatureSchemes is used to provide a list of values sent in the signature\_algorithms extension.

For more information about configuring the values of the above-mentioned parameters, see the "Ingress Gateway Microservice" and "Egress Gateway Microservice" sections in the Oracle Communications Cloud Native Core, Network Slice Selection Function Installation, Upgrade, and Fault Recovery Guide.

- 3. Save the ocnssf\_custom\_values\_25.1.100.yaml file.
- 4. Install NSSF. For more information about the installation procedure, see the *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Installation*, *Upgrade*, *and Fault Recovery Guide*.
- 5. Run Helm upgrade if you are enabling this feature after NSSF deployment. For more information about the upgrade procedure, see the *Oracle Communications Cloud Native Core, Network Slice Selection Function Installation, Upgrade, and Fault Recovery Guide.*



#### (i) Note

- NSSF does not prioritize cipher suites based on priority. To select a cipher based on priority, you must list the cipher suites in decreasing order of priority.
- NSSF does not prioritize supported groups based on priority. To select a supported group based on priority, you must list the supported group values in decreasing order of priority.
- If you want to provide values for the signature\_algorithms extension using the clientSignatureSchemes parameter, the following comma-separated values must be provided to deploy the services:
  - rsa\_pkcs1\_sha512
  - rsa\_pkcs1\_sha384
  - rsa\_pkcs1\_sha256

#### (i) Note

By default, it is null.

- The mandatory extensions as listed in RFC 8446 cannot be disabled using the clientDisabledExtension attribute on the client or using the serverDisabledExtension attribute on the server side. The following is the list of the extensions that cannot be disabled:
  - supported\_versions
  - key\_share
  - supported\_groups
  - signature\_algorithms
  - pre\_shared\_key

#### Observe

#### **Metrics**

The following metrics are available for this feature:

- oc\_ingressgateway\_incoming\_tls\_connections
- oc\_egressgateway\_outgoing\_tls\_connections
- security\_cert\_x509\_expiration\_seconds

For more information about metrics, see **NSSF Metrics** section.

#### **KPIs**

There are no new KPIs for this feature.

#### **Alerts**

The following alerts are available for this feature:

OcnssfTLSCertificateExpireMinor



- OcnssfTLSCertificateExpireMajor
- OcnssfTLSCertificateExpireCritical

#### Note

Alert gets raised for every certificate that will expire in the above time frame. For example, NSSF supports both RSA and ECDSA. So, we have configured two certificates. Accordingly, let us suppose RSA certificate is about to expire in 6 months in this situation only one alert will be raised and if both are about to expire then two alerts will be raised.

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- **1. Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Troubleshooting Guide*.
- **2.** Raise a service request: See My Oracle Support for more information on how to raise a service request.

## 4.4 Traffic Segregation

This feature provides end-to-end traffic segregation to NSSF based on traffic types. Within a Kubernetes cluster, traffic segregation can divide applications or workloads into distinct sections such as OAM, SBI, Kubernetes control traffic, etc. The Multus CNI container network interface (CNI) plugin for Kubernetes enables attaching multiple network interfaces to pods to help segregate traffic from each NSSF microservice.

This feature addresses the challenge of logically separating IP traffic of different profiles, which are typically handled through a single network (Kubernetes overlay). The new functionality ensures that critical networks are not cross-connected or sharing the same routes, thereby preventing network congestion.

With traffic segregation, operators can segregate traffic to external feeds and applications more effectively. Previously, all external traffic was routed through the same external network, but now, egress traffic from the NSSF pods can be directed through non-default networks to third-party applications. This separation is achieved by leveraging cloud-native infrastructure and the load balancing algorithms in OCCNE.

The feature supports the configuration of separate networks, Network Attachment Definitions (NADs), and the Cloud Native Load Balancer (CNLB). These configurations are crucial for enabling cloud native load balancing, facilitating ingress-egress traffic separation, and optimizing load distribution within NSSF.

#### Note

The Traffic Segregation feature is only available in NSSF if OCCNE is installed with CNLB.



#### **Cloud Native Load Balancer (CNLB)**

CNE provides Cloud Native Load Balancer (CNLB) for managing the ingress and egress network as an alternate to the existing LBVM, lb-controller, and egress-controller solutions. You can enable or disable this feature only during a fresh CNE installation. When this feature is enabled, CNE automatically uses CNLB to control ingress traffic. To manage the egress traffic, you must preconfigure the egress network details in the cnlb.ini file before installing CNE.

For more information about enabling and configuring CNLB, see *Oracle Communications Cloud Native Core, Cloud Native Environment User Guide*, and *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.

#### **Network Attachment Definitions for CNLB**

A Network Attachment Definition (NAD) is a resource used to set up a network attachment, in this case, a secondary network interface to a pod. NSSF supports two types of CNLB NADs:

#### 1. Ingress Network Attachment Definitions

Ingress NADs are used to handle inbound traffic only. This traffic enters the CNLB application through an external interface service IP address and is routed internally using interfaces within CNLB networks.

Naming Convention:nf-<service network name>-int

#### 2. Egress Only Network Attachment Definitions

Egress Only NADs enable outbound traffic only. An NF pod can initiate traffic and route it through a CNLB application, translating the source IP address to an external egress IP address. An egress NAD contains network information to create interfaces for NF pods and routes to external subnets.

#### Requirements:

- Ingress NADs are already created for the desired internal networks.
- Destination (egress) subnet addresses are known beforehand and defined under the cnlb.ini file's egress dest variable to generate NADs.
- The use of an Egress NAD on a deployment can be combined with Ingress NADs to route traffic through specific CNLB apps.
- Naming Convention:nf-<service\_network\_name>-egr

#### **Managing Ingress and Egress Traffic Segregation**

#### **Enable:**

This feature is disabled by default. To enable this feature, you must configure the network attachment annotations in the ocnssf custom values 25.1.100.yaml file.

#### Configuration

For more information about Traffic Segregation configuration, see " Configuring Traffic Segregation" section in *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Installation*, *Upgrade*, *and Fault Recovery Guide*..

#### Observe

There are no Metrics, KPIs, or Alerts available for this feature.



#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- Collect the logs: For more information on how to collect logs, see Oracle Communications Cloud Native Core, Network Slice Selection Function Troubleshooting Guide.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

## 4.5 Support for Common Service APIs in CNC Console

The configuration for common service APIs was supported only using REST. With the implementation of this feature, NSSF now supports the configuration of Ingress Gateway and Egress Gateway parameters using the CNC Console. You can perform HTTP methods such as GET and PUT using the Console.

For more information about the common service APIs, see "Common Services REST APIs" section in the *Oracle Communication Cloud Native Core, Network Slice Selection Function REST Specification Guide*.

#### Managing common service APIs in the CNC Console

#### **Enable**

This feature is enabled automatically along with the NRF instance deployment.

#### Configure

You can configure the common services APIs in the CNC Console. For more information, see <a href="Common Services Configuration">Common Services Configuration</a> section.

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

## 4.6 Enhanced Computation of AllowedNSSAI in NSSF

As per 3GPP specification, NSSF utilizes AMF (Access and Mobility Management Function) to manage the slice selection criteria. Here, if a newly configured NSSAI is not supported by the Radio Access Network (RAN), the AMF designates it as the Rejected-NSSAI. To allow the newly configured NSSAI, UE needs to request for a re-registration. It allows the AMF to disregard a UE's Requested NSSAI and formulate the AllowedNSSAI based on subscription details and support from the RAN

The objective of this feature is to maintain a consistent approach to handling slice subscription changes between AMF and NSSF. Recognizing the need for adaptive and subscriber centric configurations of NSSAI, this feature allows NSSF to generate AllowedNSSAI based on individual user equipment (UE) subscriptions and Tracking Area characteristics.



When this feature is enabled, NSSF generates AllowedNSSAI based on User Equipment (UE) subscription details and the supported configuration within the Tracking Area. Conversely, when the feature is disabled, the NSSF adheres to the 3GPP specification 29.531, constructing AllowedNSSAI based on the intersection of Requested and Subscribed NSSAI and also operator policy.

#### NSSAI Configuration (When the feature is enabled):

- Subscriber-Driven Configuration: The NSSF will dynamically build AllowedNSSAI based on the UE subscription information. This ensures that network slices are aligned with the specific requirements and preferences of individual subscribers.
- Tracking Area Considerations: The NSSF takes into account the supported configuration
  within the Tracking Area, optimizing the network slice selection based on the geographical
  location and network capabilities in real time.

#### Static NSSAI Configuration (When the feature is disabled):

- Compliance with 3GPP Specification 29.531: When the feature is disabled, the NSSF follows the guidelines outlined in the 3GPP specification 29.531. This involves creating AllowedNSSAI based on the intersection of Requested and Subscribed NSSAI, providing a standardized approach to network slice configuration.
- Maintaining Compatibility: Disabling the dynamic NSSAI configuration ensures compatibility with industry standards and facilitates interoperability across different network elements and vendors.

#### Managing Enhanced Computation of AllowedNSSAI in NSSF

#### **Enable:**

You can enable this feature using REST API or CNC Console.

#### **Enable using REST API**

- Use the following API path: {apiRoot}/nnsf-configuration/v1/nssfSystemOptions/
- 2. To enable the feature for a specific PLMN, set the EnableEnhancedAllowedNSSAIComputation parameter to **true**. The default value for this parameter is false.
- Run GET service method to get the EnableEnhancedAllowedNSSAIComputation value for the PLMN.
- Run PUT service method to update the EnableEnhancedAllowedNSSAIComputation value for the PLMN.

For more information about API path, see "NssfSystemOptions" section in *Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide*.

#### **Enable using CNC Console**

For more information, see **NSSF System Option** section.

#### Observe

#### Metrics

There are no new metrics for this feature.

#### **KPIs**



There are no new KPIs for this feature.

#### **Alerts**

There are no alerts generated for this feature.

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

## 4.7 LCI and OCI Headers

Within the complex 5G architecture, network overload scenarios are common. The exchanges of data between producer and consumer Network Functions (NFs) often involve significant message and notification volumes, necessitating a precise approach to load balancing. This is imperative to prevent network failures triggered by overload conditions.

In such scenarios, it becomes crucial for consumer NFs to be promptly notified when the producer NF approaches an overloaded state. This enables consumer NFs to implement corrective actions proactively.

To address these challenges, the introduction of LCI and OCI Headers plays a pivotal role in optimizing communication between NSSF and its consumer NFs. They provide consumer NFs with real-time insights into the operational status of the NSSF resources, facilitating efficient traffic management.

These headers provide essential load and overload information for consumer NFs to optimize traffic distribution and take proactive measures during network overload scenarios.

The headers are integrated into outgoing responses, based on load levels at the Ingress Gateway. They server as communication tools and allow Network Functions (NFs) to share crucial load information, ensuring an architecture where 5G Core Network remains stable and high performing even during heavy load conditions.

#### **LCI** Header

The LCI header comprises overall load related information such as the timestamp of load data generation, the current load of the NF, and the scope of the load information. For example, there are two LCI headers:

- NF scope LCI header
- NF-service scope LCI header

#### **Examples of LCI Headers**

#### **NF Scope LCI Header:**

3gpp-sbi-lci: Timestamp: "Tue, 19 Sep 2023 13:47:41 UTC"; Load-Metric: 1%; NF-Instance: 9faf1bbc-6e4a-4454-a507-aef01a101a01

#### Service Scope LCI Header:

3gpp-sbi-lci: Timestamp: "Mon, 25 Sep 2023 11:30:17 UTC"; Load-Metric: 0%; NF-Service-Instance: ae870316-384d-458a-bd45-025c9e748976



#### **OCI Header**

The OCI header communicates information about overload conditions. This information encompasses the timestamp when the overload condition was detected, Overload-Reduction-Metric, and Overload Validity period.

#### **Examples of OCI Headers**

#### **NF Scope OCI Header:**

3gpp-Sbi-Oci:Timestamp: "Mon, 02 May 2022 07:43:48 UTC"; Period-of-Validity: 30s; Overload-Reduction-Metric: 5.0%; NF-Instance: 5a7bd676-ceeb-44bb-95e0-f6a55a328b03

#### Service Scope OCI Header:

3gpp-Sbi-Oci:Timestamp: "Mon, 02 May 2022 07:43:48 UTC"; Period-of-Validity: 30s; Overload-Reduction-Metric: 5.0%; NF-Service-Instance: 5a7bd676-ceeb-44bb-95e0-f6a55a328b03

Both the LCI and OCI headers are incorporated in HTTP response messages without triggering additional signaling, ensuring a more efficient communication process. Here is how they help:

- The LCI header conveys the overall load of an NF, assisting in decisions regarding the acceptance or rejection of new requests to prevent further overload.
- In contrast, the OCI header communicates specific overload conditions, helping NFs take informed actions to mitigate these conditions.
- The LCI and OCI headers complement each other, allowing an NF to reduce the number of requests it sends to other NFs in response to an OCI header, even if it's not yet overloaded.
- This proactive measure prevents overload conditions from replicating.

#### **Use Cases**

- As of now, LCI and OCI Headers are supported at the Ingress Gateway only; not at the Egress Gateway.
- NSSF can utilize the LCI header to share its load information with the AMF (Access and Mobility Management Function). This information allows the AMF to make informed decisions about directing new User Equipment (UE) connections to the NSSF.
- Conversely, the NSSF can employ the OCI header to notify the Radio Access Network (RAN) of overload conditions, enabling the AMF to reduce the number of UEs it routes to NSSF.

#### **Future Enhancements**

Currently, this feature supports LCI-OCI headers at Ingress Gateway. The support for LCI and OCI headers in outgoing messages from Egress Gateway will be enabled in the future.

#### Managing LCI and OCI Headers

#### **Enable:**

You can enable LCI and OCI Headers by performing the following Helm configurations globally and at the Ingress Gateway:



#### Global Helm Configuration

Enable: You can enable LCI and OCI Headers globally at the Network Function (NF) level by setting the values of nssflciEnabled and nssfOciEnabled parameters as true, respectively.

#### Ingress Gateway Helm Configuration

- Enable: You can enable LCI and OCI Headers globally at Ingress Gateway level by setting the lciHeaderConfig.enabled and ociHeaderConfig.enabled parameters as true, respectively.
- Configure: You can configure LCI and OCI Headers at Ingress Gateway using the Helm based configuration:

```
## This is mandatory for LCI and OCI feature as this is required by
perf-info service to get the load information of the services from
prometheus
perf-info:
  configmapPerformance:
     prometheus: http://occne-prometheus-server.occne-infra
ingress-gateway:
 #To remove the Producer header from Ingress Response when LCI is
enabled
 globalRemoveResponseHeader:
  - name: *producer
  # ******
             Sub-Section Start: LCI/OCI Ingress Gateway Parameters
#**********************
  # Engineering Parameter Start
 global:
   lciHeaderConfig:
     enabled: *lcienable
     # difference between previous threshold and current threshold for
lci header will be added when the difference crosse the mentioned value
     loadThreshold: 30
     # Validity period after which lci header will be added to reponse
header if delta of threshold is not breached
     localLciHeaderValidity: 60000 #(value in milliseconds)
     ## This header needs to be same which is being sent along with
request in microservice
     producerSvcIdHeader: *producer
```



```
ociHeaderConfig:
      enabled: *ocienable
      ## This header needs to be same which is being sent along with
reuest microservice
      producerSvcIdHeader: *producer
      validityPeriod: 10000 #(value in milliseconds)
      ## The range of the cpu load for which the ingress gateway will
get notified regarding the criticality of the load.
      overloadConfigRange: #Note - minor, major and critical conditions
should cover complete range of 0 to 100 both inclusive for it to be a
valid config
       minor: "[75-80]"
       major: "[81-90]"
       critical: "[91-100]"
      ## The range of the cpu load which needs to be decreased from the
consumer when a particular criticality has reached.
     reductionMetrics:
       minor: 5 #(Possible values 1 to 9 both inclusive)
       major: 15 #(Possible values 5 to 15 both inclusive)
        critical: 25 #(Possible values 10 to 50 both inclusive)
    nfInstanceId: "9faf1bbc-6e4a-4454-a507-aef01a101a01"
      ## This is a mapping for service name to service instance id for
which service the LCI and OCI headers needs to be enabled. Default
values are given with 'ocnssf' as release name. It must be configured
by operator in case release name is changed.
      ## only nsselection and nsavailability services should be
configured. As these are the two services for which LCI/OCI headers is
supported.
      ## Format is <releaseName>-<serviceName> . Eg: if release name is
given 'ocnssf-test' then svc name should be 'ocnssf-test-nsselection'
and 'ocnssf-test-nsavailability'
   svcToSvcInstanceIdMapping:
      - svcName: ocnssf-nsselection
        serviceInstanceId: "ae870316-384d-458a-bd45-025c9e748976"
      - svcName: ocnssf-nsavailability
       serviceInstanceId: "ae870316-384d-458a-bd45-025c9e748996"
   perfInfoConfig:
      ## the interval when perf-info will fetch the cpu load from
prometheus
      pollingInterval: 5000 #(value in milliseconds)
      serviceName: "ocnssf-perf-info"
     port: 5905
      perfInfoRequestMap: "/load"
  # Engineering Parameter End
  # ****** Sub-Section End: LCI/OCI Ingress Gateway Parameters
******
  # ****** Sub-Section Start: DB credentials Ingress Gateway
Parameters ********
#***********************
  dbConfig:
```



For more information about the Helm parameters, see "Customizing NSSF" section in *Oracle Communications Cloud Native Core, Network Slice Selection Function Installation, Upgrade, and Fault Recovery Guide.* 

#### Observe

#### **Metrics**

There are no new metrics for this feature.

#### **KPIs**

There are no new KPIs for this feature.

#### Alerts

There are no alerts generated for this feature.

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Troubleshooting Guide*.
- **2.** Raise a service request: See My Oracle Support for more information on how to raise a service request.

## 4.8 Server Header in NSSF

One of the core functionalities of the Network Slice Selection Function (NSSF) is to manage the security aspects of the 5G network. As part of this role, NSSF handles various requests from other network functions (NFs) and network entities over the HTTP protocol. On receiving these requests, NSSF validates and processes them before issuing responses to the requesting NFs or network entities.

In such scenarios, NFs and other network entities may encounter issues resulting in error responses. It becomes imperative for consumer NFs to pinpoint the source of the error, so they



can undertake troubleshooting and corrective measures. The integration of this feature at NSSF helps to determine the originator of the error response.

This feature offers the support for Server Header in NSSF responses, which contain crucial information about the origin of an error response and the type of the error encountered. Thus, the Server Header enhances the behavior of NSSF while responding to requests, particularly the error responses.

#### Note

This feature is applicable in scenarios where NSSF generates error responses. It does not affect normal response behavior.

A Server Header starts with the value of NF Type, followed by a "-" and any other specific information, if needed, afterward. It is expected to be present in all NSSF responses in the following format:

```
<NF Type>-<Instance-Id>
Where,
```

- <NF Type> is the type of the NF.
- <NF Instance-Id> is the unique identifier of the NF instance generating the response.

For example, the following combinations are applicable to NSSF:

```
NSSF-<NSSF's Instance-Id>
```

#### Where,

- NSSF is the <NF Type>.
- <NSSF's Instance-Id> is the unique identifier of the NSSF instance generating the response.

#### **Managing Server Header in NSSF**

#### **Enable:**

You can enable this feature using REST API or CNC Console.

#### **Enable using REST API**

By default, this feature is disabled. To enable it, REST API needs to be invoked and the enabled flag needs to be updated to true in the following URI:

/{nfType}/nf-common-component/v1/{serviceName}/serverheaderdetails

#### **Example**

#### **Example of Request or Response Body to Enable Server Header:**

```
"enabled": true,
"errorCodeSeriesId": "E1",
"configuration": {
  "nfType": "NSSF",
 "nfInstanceId": "9faf1bbc-6e4a-4454-a507-aef01a101a01"
```



```
}
}
```

#### **Example of Request or Response Body to Disable Server Header:**

```
{
  "enabled": false,
  "errorCodeSeriesId": "E1",
  "configuration": {
     "nfType": "NSSF",
     "nfInstanceId": "9faf1bbc-6e4a-4454-a507-aef01a101a01"
  }
}
```

#### (i) Note

- enabled is used to enable or disable the feature.
- nfType and nfInstanceId are used to form Server Header.
- In the mentioned configuration, when sending a response to AMF, the Server Header will be appended by the NSSF with the value "NSSF-9faf1bbc-6e4a-4454-a507-aef01a101a01"
- The values in the above example are samples. Ensure that you update the values
  of the following parameters according to your deployment:
  - nfType must be NSSF.
  - errorCodeSeriesId: A valid configured value.
  - nfInstanceId: NSSF's valid instance value. It must be same as NSSF's instance ID.

#### **Enable using CNC Console**

For more information, see Server Header Details.

#### Configure

Perform the REST API or CNC Console configurations in the following sequence to configure this feature:

- 1. Configure **errorcodeserieslist** to update the **errorcodeserieslist** that are used to list the configurable exception or error for an error scenario in Ingress Gateway.
- 2. Configure **serverheaderdetails** to enable the feature.
- <Optional>Configure routesconfiguration to map route ID and its corresponding routelevel configuration.

#### (i) Note

If this configuration is done for Server Header, then for this particular route, it will take precedence over the **serverheaderdetails** configuration.



#### Note

- If routesconfiguration is done without errorCodeSeriesId then errorCodeSeriesId configured at serverheaderdetails is picked up, if Server Header is enabled there.
- If Server Header is enabled without configuring **errorCodeSeriesId** then it will be applied for all error codes. This is not a recommended configuration.

For more details about REST APIs, see "REST API Configurations for Server Header Feature" in Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.

#### Observe

#### Metrics

There are no new metrics for this feature.

#### **KPIs**

There are no new KPIs for this feature.

#### **Alerts**

There are no alerts generated for this feature.

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- Collect the logs: For more information on how to collect logs, see Oracle Communications Cloud Native Core, Network Slice Selection Function Troubleshooting Guide.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

## 4.9 Support for User-Agent Header

In 5G networks, producer Network Functions (NFs) cannot identify or validate a consumer on their own. To overcome this, 3GPP has introduced User-Agent headers, which are added to consumer service requests. This field is included in the HTTP (Hypertext Transfer Protocol) request that a consumer sends to the producer to identify itself and provide information about the NF making the request.

This feature enables the usage of the User-Agent Header in NSSF.

NSSF support the following inter-NF communication and service request functionalities:

- NSSF sends notifications to AMF.
- NSSF sends registration and heartbeat request to NRF.

This enhancement enables NSSF to include the User-Agent Header in every HTTP/2 request that it sends over any Service Based Interface (SBI) to a producer NF (for example, AMF and NRF). The User-Agent Header in NSSF's HTTP/2 requests helps a producer NF to identify NF type of client that has sent a request. Here, it helps:

AMF in identifying the NSSF that sent the notification.



NRF in identifying the NSSF that sent the subscription, registration, or heartbeat request to NRF.

#### Structure of an User-Agent Header

An User-Agent Header starts with the value of NF type, followed by a "-" and any other specific information, if needed afterwards. It is expected to be present in all the service requests and notification in the following formats:

- <NF Type>
- <NF Type>-<Instance-Id>
- <NF Type>-<Instance-Id> <FODN>

#### Where,

- <NF Type> is the type of the NF.
- <Instance-Id> is the instance ID of the NF.
- <FQDN> is the FQDN of the NF.

For example: The following combinations are applicable to NSSF:

**NSSF** 

NSSF-<NSSF's Instance-Id>

NSSF-<NSSF's Instance-Id> <NSSF's FQDN>

When the User-Agent Header is not included in the incoming requests sent to AMF or NRF, the corresponding metric cannot gather information about the origin of the service request. Nevertheless, the request is still processed successfully without any problems, but the AMF or NRF are not able to identify the NSSF from which the request has originated.



#### (i) Note

The onus is on operator to configure the values correctly as defined in the syntax explained above.

#### Managing the Support for User-Agent Header

#### **Enable:**

You can enable this feature using REST API or CNC Console.

#### **Enable using REST API**

- Use the following API path: /{nfType}/nf-common-component/v1/{serviceName}/useragentheader
- Set enabled as true.
- Run the API using PUT method with the proposed values given in the Rest API. For more information about API path, see "Configurations to Enable or Disable User-Agent Header" section of "Egress Gateway REST APIs" in Oracle Communications Cloud Native Core. Network Slice Selection Function REST Specification Guide.

```
Given below is a sample REST API configuration to enable this feature:
```

```
"enabled": true,
```



```
"nfType": "NSSF",
    "nfInstanceId": "9faf1bbc-6e4a-4454-a507-aef01a101a01",
"nfFqdn": "nssf.oracle.com",
"addFqdnToHeader": true,
    "overwriteHeader": true
}
```

#### (i) Note

- In the mentioned configuration, when sending notifications to AMF, the User-Agent Header will be appended by the NSSF with the value NSSF-9faf1bbc-6e4a-4454-a507-aef01a101a01 nssf.oracle.com.
- The nfInstanceId and nfFqdn values in the above example are samples. Ensure
  that you update the values of the nfInstanceId and nfFqdn parameters
  accordingly.

#### **Enable using CNC Console**

For more information, see User Agent Header Generation.

#### Observe

#### **Metrics**

The following metric is used to provide information about this feature:

 oc\_egressgateway\_user\_agent\_consumer\_total: This metric is applicable whenever the feature is enabled and User-Agent Header is getting generated.

For information about the metrics, see Egress Gateway Metrics.

#### **Alerts**

There are no alerts generated for this feature.

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

## 4.10 Ingress Gateway Pod Protection

This feature protects the Ingress Gateway pods from overloading due to uneven traffic distribution, traffic bursts, or congestion. During overload conditions, the Ingress Gateway pods may undergo stability issues. As a front end microservice for HTTP traffic, it is important for Ingress Gateway to have pod protection implemented.

The pod protection is performed based on the CPU consumption of the Ingress Gateway Pods as explained in the <u>Congestion State Parameters</u>. It is measured at different load states mentioned in the <u>Ingress Gateway Load States</u>.





#### (i) Note

Horizontal POD Autoscaling (HPA) at Ingress Gateway microservice and pod protection mechanism at Ingress Gateway microservice are independent features with different trigger conditions. While HPA considers microservice load, pod protection mechanism works only on the POD load. Therefore, their order of triggering cannot be predicted.

In a service mesh based deployment, all incoming connections to the pod get terminated at the sidecar container, then the sidecar container creates a new connection toward the application container. These incoming connections from the peer are managed by the sidecar and outside the purview of the application container.

Hence when the Ingress Gateway container reaches DOC or Congested level, in a service mesh based deployment, the Ingress Gateway container will only be able to stop accepting new connections from the sidecar container. Also in this state, the Ingress Gateway container will reduce the concurrency of the existing connections between the sidecar container and the Ingress Gateway container. Any new request received over a new connection may get accepted or rejected based on the sidecar connection management.

In a non-service mesh based deployment, all incoming connections to the pod get terminated at the Ingress Gateway container. Hence when the Ingress Gateway container reaches DOC or Congested level, the Ingress Gateway container will stop accepting new connections. Also in this state, the Ingress Gateway container will reduce the concurrency of the existing connections between the peer and the Ingress Gateway container. Any new request received over a new connection will result in to a request timeout at the peer.

#### **Congestion State Parameters**

In the Pod Protection feature, each Ingress Gateway microservice pod monitors its congestion state. This state is tracked in terms of CPU consumption, measured in nanoseconds, using Kubernetes cgroup (cpuacct.usage).

It is periodically monitored and calculated using the following formula. Then, it is compared against the CPU thresholds configured through the Rest API to determine the congestion state. For more information about the parameters, see Oracle Communications Cloud Native Core. Network Slice Selection Function REST Specification Guide.

Where,

CurrentCpuUsage is the counter reading at current periodic cycle.

LastCpuUsage is the counter reading at previous periodic cycle.

CurrentTime is the current time snapshot.

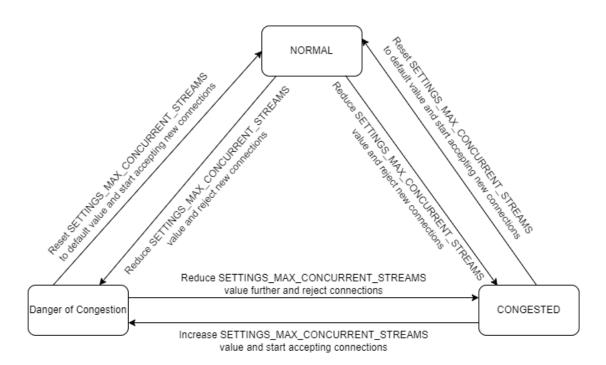
LastSampletime is the previous periodic cycle time snapshot.

CPUs is the total number of CPUs for a given pod.



#### **Ingress Gateway Load States**

The following states are used to detect overload conditions. This ensures the protection and mitigation of pods from entering an overload condition, while also facilitating necessary actions for recovery.



#### (i) Note

The transition can occur between any states based on the congestion parameters. The threshold for these congestion parameters is preconfigured and must not be changed.

- Congested State: This is the upper bound state where the pod is congested. This means
  one or more congestion parameters are above the configured thresholds for the congested
  state. For more information about the configuration using REST API, see Oracle
  Communications Cloud Native Core, Network Repository Function REST Specification
  Guide. The pod can be transitioned to the Congested State either from the Normal State or
  the DoC state. When the pod reaches this state, the following actions are performed:
  - new incoming HTTP2 connection requests are not accepted.
  - the pod gradually decrements the number of concurrent streams by updating SETTINGS\_MAX\_CONCURRENT\_STREAMS parameter in a SETTINGS frame to the configured maxConcurrentStreamsPerCon value at a regular interval. The concurrent streams are decremented based on the value configured in decrementBy parameter. And, the regular interval is configured in the decrementSamplingPeriod parameter.
- Danger of Congestion (DOC): This is the intermediate state where the pod is approaching a congested state. This means if CPU is above the configured thresholds for the DoC state.
  - any new incoming HTTP2 connection requests are not accepted.



- if the pod is transitioning from the Normal State to the DoC state, the pod gradually decrements the number of concurrent streams by updating SETTINGS\_MAX\_CONCURRENT\_STREAMS parameter in a SETTINGS frame to the configured maxConcurrentStreamsPerCon value at a regular interval. The concurrent streams are decremented based on the value configured in decrementBy parameter. And, the regular interval is configured in the decrementSamplingPeriod parameter.
- if the pod is transitioning from the Congested State to the DoC state, the pod gradually increments the number of concurrent streams by updating SETTINGS\_MAX\_CONCURRENT\_STREAMS parameter in a SETTINGS frame to the configured maxConcurrentStreamsPerCon value at a regular interval. The concurrent streams are incremented based on the value configured in incrementBy parameter. And, the regular interval is configured in the incrementSamplingPeriod parameter.
- Normal State: This is the lower bound state where all the congestion parameters for the
  pod are below the configured thresholds for DoC and Congested states. When the pod
  reaches this state, the following actions are performed:
  - the pod will continue accepting new incoming HTTP2 connection requests.
  - the pod will continue accepting requests on the existing HTTP2 connections.
  - in case the pod is transitioning from the Congested or DoC state to Normal state, the pod gradually increments the number of concurrent streams by updating SETTINGS\_MAX\_CONCURRENT\_STREAMS parameter in a SETTINGS frame to the configured maxConcurrentStreamsPerCon value at a regular interval. The concurrent streams are incremented based on the value configured in incrementBy parameter. And, the regular interval is configured in the incrementSamplingPeriod parameter.

To avoid toggling between these states due to traffic pattern, it is required for the pod to be in a particular state for a given period before transitioning to another state. The below configurations are used to define the period till which the pod has to be in a particular state:

- stateChangeSampleCount
- monitoringInterval

Formula for calculating the period is as follows:

(stateChangeSampleCount \* monitoringInterval)

For more information about the configuration using REST API, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function REST Specification Guide*.

#### **Managing Ingress Gateway Pod Protection**

This section explains the procedure to enable and configure the feature.

#### **Enable:**

You can enable this feature using REST API or CNC Console.

#### **Enable using REST API**

Perform the REST API configurations as explained below:

- 1. Use the API path as {apiRoot}/nf-common-component/v1/{serviceName}/podprotection.
- Set enabled as true.
- 3. Set congestionControl.enabled to true.
- Run the API using PUT method with the proposed values given in the Rest API. For more information about API path, see "Configurations to enable Ingress Gateway Pod



Protection" section of "Ingress Gateway REST APIs" in *Oracle Communications Cloud Native Core*, *Network Slice Selection Function REST Specification Guide*. Given below is a sample REST API configuration to enable this feature:

```
"enabled": true,
"monitoringInterval": 100,
"congestionControl": {
    "enabled": true,
    "stateChangeSampleCount": 10,
    "actionSamplingPeriod": 3,
    "states": [
            "name": "Normal",
            "weight": 0,
            "entryAction": [
                    "action": "MaxConcurrentStreamsUpdate",
                    "arguments": {
                        "incrementBy": 30,
                         "incrementByActionSamplingPeriod": 3,
                         "maxConcurrentStreamsPerCon": 100
                },
                    "action": "AcceptIncomingConnections",
                    "arguments": {
                         "accept": true
                }
            ]
            "name": "DoC",
            "weight": 1,
            "resourceThreshold": {
                "cpu": 60,
                "memory": 60,
                "pendingMessage": 5000
            "entryAction": [
                    "action": "AcceptIncomingConnections",
                    "arguments": {
                         "accept": false
                },
                    "action": "MaxConcurrentStreamsUpdate",
                    "arguments": {
                        "incrementBy": 30,
                         "incrementByActionSamplingPeriod": 3,
                         "decrementBy": 30,
                         "decrementByActionSamplingPeriod": 1,
                         "maxConcurrentStreamsPerCon": 50
```



```
1
        },
            "name": "Congested",
             "weight": 2,
             "resourceThreshold": {
                 "cpu": 75,
                 "memory": 75,
                 "pendingMessage": 7000
            },
             "entryAction": [
                     "action": "AcceptIncomingConnections",
                     "arguments": {
                         "accept": false
                 },
                     "action": "MaxConcurrentStreamsUpdate",
                     "arguments": {
                         "decrementBy": 30,
                         "decrementByActionSamplingPeriod": 1,
                         "maxConcurrentStreamsPerCon": 5
                 }
            ]
    ]
}
```

#### **Enable using CNC Console**

For more information, see **Pod Protection**.

#### **Observe**

#### **Metrics**

The following metrics are used to provide information about this feature:

- oc\_ingressgateway\_pod\_congestion\_state: It is used to track congestion state of a pod.
- oc\_ingressgateway\_pod\_resource\_stress: It tracks CPU, memory, and queue usage (as percentages) to determine the congestion state of the POD that is performing the calculations.
- oc\_ingressgateway\_pod\_resource\_state: It tracks the congestion state of individual resources, which is calculated based on their usage and the configured threshold.
- oc\_ingressgateway\_incoming\_pod\_connections\_rejected\_total: It tracks the number of connections dropped in the congested or Danger Of Congestion (DOC) state.

For information about the metrics, see <u>Ingress Gateway Metrics</u>.

#### **Alerts**

The following alerts generated for this feature:



- OcnssfIngressGatewayPodCongestionStateWarning
- OcnssfIngressGatewayPodCongestionStateMajor
- OcnssfIngressGatewayPodResourceStateWarning
- OcnssfIngressGatewayPodResourceStateMajor

For more information about alerts, see NSSF Alerts.

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- **1. Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

## 4.11 Monitoring the Availability of SCPs using SCP Health APIs

With the introduction of this feature, NSSF determines the availability and reachability status of all the SCPs configured either statically by the operator or through <a href="DNS SRV">DNS SRV</a> based selection of the SCP sets. With this feature, NRF determines the availability and reachability status of all SCPs irrespective of the configuration types. This feature is an enhancement to the existing SBI routing functionality. Egress Gateway microservice interacts with SCP on their health API endpoints using HTTP2 OPTIONS method. It monitors the health of configured SCP peers to ensure that the traffic is routed directly to the healthy peers. This enhancement avoids routing or rerouting towards unhealthy peers, thus minimizing the latency time.

Egress Gateway microservice maintains the health status of all available and unavailable SCPs. It maintains the latest health of SCPs by periodically monitoring and uses this data to route egress traffic to the healthy SCP.

#### (i) Note

- This is not a standalone feature but an add-on to the existing SBI Routing feature, which means this feature is activated only if the SBI Routing feature is enabled.
- Health monitoring can only be enabled for the peers which belong to a peerset associated with a SBI Routing filter.

#### Managing Monitoring the Availability of SCPs using SCP Health APIs

#### **Prerequisites**

During the installation, peermonitoringconfiguration is set to false by default. Since this feature is an add-on to the existing SBI Routing feature, it will be activated if the sbirouteconfig is enabled. To enable this feature, perform the following:

#### **Configure Using REST API**

You can also enable this feature using the REST API configurations at Egress Gateway in the following sequence:



Configure peerconfiguration to define the list of peers to which Egress Gateway can send request.



#### (i) Note

peerconfiguration must consist of healthApiPath even though peermonitoringconfiguration is set to false by default. Configure virtualHost under peerconfiguration where the AMF query is sent.

Here is a sample configuration:

#### **PUT Request**

```
curl -v -X PUT "http://{{host}}}:{{port}}/nssf/nf-common-component/v1/egw/
peerconfiguration" -H "Content-Type: application/json" --data-raw
'[{"id": "peer1", "host": "scp1", "port": "8080", "apiPrefix":
"/","healthApiPath":"/health/v1"},{"id": "peer2","host": "scp2","port":
"8080", "apiPrefix": "/", "healthApiPath": "/health/v2"}, { "id":
"peer3", "host": "scp3", "port": "8080", "apiPrefix": "/", "healthApiPath": "/
health/v3"},{"id": "peer4","host": "scp4","port": "8080","apiPrefix":
"/", "healthApiPath": "/health/v4"}, { "id": "peer5", "virtualHost":
"xyz.test.com", "apiPrefix": "/", "healthApiPath": "/health/v5"}, { "id":
"peer6", "virtualHost": "abc.test.com", "apiPrefix": "/", "healthApiPath": "/
health/v6"}]'
```

2. Configure peersetconfiguration to logically group the peers into sets.

### ① Note

- peerIdentifier must be the value of SCP peer configured in peerConfiguration.
- You cannot configure multiple virtual hosts as peers in the same peer set.
- Configure the priority for each SCP peer in the set. Depending on the priority, it selects the primary, secondary, or tertiary SCP peers to route requests.

Here is a sample configuration:

#### **PUT Request**

```
curl -v --http2-prior-knowledge -X PUT "http://{{host}}:{{port}}/nssf/nf-
common-component/v1/egw/peersetconfiguration" -H "Content-Type:
application/json" -d '[{"id":"set0","httpConfiguration":[{"priority":
1, "peerIdentifier": "peer1" }, { "priority": 2, "peerIdentifier": "peer2" },
{"priority": 3, "peerIdentifier": "peer3"}, { "priority": 4, "peerIdentifier":
"peer4"}],"httpsConfiguration":[{"priority": 1,"peerIdentifier": "peer1"},
{"priority": 2, "peerIdentifier": "peer2"}, { "priority": 3, "peerIdentifier":
"peer3"}, { "priority": 4, "peerIdentifier": "peer4" } ] },
{"id":"set1","httpConfiguration":[{"priority": 1,"peerIdentifier":
"peer5"}], "httpsConfiguration":[{"priority": 1, "peerIdentifier":
"peer6"}]}]'
```



3. Configure or update errorcriteriasets. Here is a sample configuration:

#### **PUT Request**

```
curl -v --http2-prior-knowledge -X PUT "http://{{host}}:{{port}}/nssf/nf-
common-component/v1/egw/sbiroutingerrorcriteriasets" -H "Content-Type:
application/json" -d '[{"id":"criteria_0","method":
["GET","POST","PUT","DELETE","PATCH"],"exceptions":
["java.util.concurrent.TimeoutException","java.net.UnknownHostException"]},
{"id":"criteria_1","method":
["GET","POST","PUT","DELETE","PATCH"],"response":{"statuses":
[{"statusSeries":"4xx","status":[400,404]},{"statusSeries":"5xx","status":
[500,503]}]}]
```

**4.** Configure or update erroractionsets. Here is a sample configuration:

#### **PUT Request**

```
curl -v --http2-prior-knowledge -X PUT "http://{{host}}:{{port}}/nssf/nf-
common-component/v1/egw/sbiroutingerroractionsets" -H "Content-Type:
application/json" -d
'[{"id":"action_0","action":"reroute","attempts":3,"blacklist":
{"enabled":false,"duration":60000}},
{"id":"action_1","action":"reroute","attempts":3,"blacklist":
{"enabled":false,"duration":60000}}]'
```

**5.** Configure routesconfiguration to define the route and reroute parameters.

#### (i) Note

- The configuration under sbiRoutingConfiguration corresponds to the SBI-Routing specific configuration.
- If SBIRouting functionality is required, then configure SBIRoutingFilter. If reroute mechanism is required for that route, then configure SBIReroute filter with retries, methods, and statuses.
- peerSetIdentifier must be the value configured during peersetconfiguration.

Here is a sample configuration:

#### **PUT Request**

```
curl -v --http2-prior-knowledge -X PUT "http://{{host}}:{{port}}/nssf/nf-
common-component/v1/egw/routesconfiguration" -H "Content-Type:
application/json" -d '[{"id":"egress_scp_proxy1","uri":"http://
localhost:32068/","order":0,"metadata":
{"httpsTargetOnly":false,"httpRuriOnly":false,"sbiRoutingEnabled":true},"pr
edicates":[{"args":{"pattern":"/notification/
amf2/"},"name":"Path"}],"filters":[{"name":"SbiRouting","args":
{"peerSetIdentifier":"set0","customPeerSelectorEnabled":true,"errorHandling
":[{"errorCriteriaSet":"criteria_1","actionSet":"action_1","priority":1},
{"errorCriteriaSet":"criteria 0","actionSet":"action 0","priority":2}]}}]}],
```



```
{"id": "default route", "uri": "egress://request.uri", "order":
100, "filters": [{"name": "DefaultRouteRetry"}], "predicates": [{"args":
{"pattern": "/**"}, "name": "Path"}]}]'
```

Set the value of enabled under sbiRoutingConfiguration to true to route the AMF queries through SCP configured in the id attribute.

#### (i) Note

peerconfiguration and peersetconfiguration can be either set to empty list or populated with values. These attributes are used for routing only if sbiRoutingConfiguration is enabled for a particular route.

7. After above configurations, configure enable in peermonitoring configuration as true to enable peer monitoring. By default, enable is set to false.

#### (i) Note

- Peer Monitoring can be enabled to use this feature, where Egress Gateway dynamically monitors the health of the peers configured.
- It is mandatory to configure peerconfiguration with healthApiPath if peermonitoringconfiguration is enabled.

Here is a sample configuration:

#### **PUT Request**

```
curl -v --http2-prior-knowledge -X PUT "http://{{host}}:{{port}}/nssf/nf-
common-component/v1/egw/peermonitoringconfiguration" -H "Content-Type:
application/json" --data-raw ' {"enabled": true, "timeout":
1000, "frequency": 2000, "failureThreshold": 3, "successThreshold": 3}'
```

#### (i) Note

The IPs and parameter values in the examples are just placeholders. Replace them with your own settings for the cURLs to function correctly.

For detailed information about the REST APIs and parameters, see "Egress Gateway REST APIs" in Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.

#### **Configure Using CNC Console**

Taking into account the prerequisites criteria and recommended sequence explained in sections above, you can refer to the Egress Gateway Configurations section in Configuring NSSF using CNC Console for configuring this feature using CNC Console.

#### **Observe**

#### **Metrics**



The following metrics are used to provide information about this feature:

- oc\_egressgateway\_peer\_health\_status
- oc\_egressgateway\_peer\_health\_ping\_request\_total
- oc\_egressgateway\_peer\_health\_ping\_response\_total
- oc\_egressgateway\_peer\_health\_status\_transitions\_total
- oc\_egressgateway\_peer\_count
- oc\_egressgateway\_peer\_available\_count

For information about the metrics, see NSSF Metrics.

#### **Alerts**

The following alerts are applicable for this feature:

- OcnssfScpMarkedAsUnavailable
- OcnssfAllScpMarkedAsUnavailable

For more information about the alerts, see NSSF Alerts.

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

## 4.12 Support for Kubernetes Resource

### 4.12.1 Network Policies

Network Policies are an application-centric construct that allows you to specify how a pod is allowed to communicate with various network entities. To control communication between the cluster's pods and services and to determine which pods and services can access one another inside the cluster, it creates pod-level rules.

Previously, NSSF had the privilege to communicate with other namespaces, and pods of one namespace could communicate with others without any restriction. Now, namespace-level isolation is provided for the NSSF pods, and some scope of communications is allowed between the NSSF and pods outside the cluster. The network policies enforces access restrictions for all the applicable data flows except communication from Kubernetes node to pod for invoking container probe.

#### **Managing Support for Network Policies**

#### **Enable**

To use this feature, network policies need to be applied to the namespace in which NSSF is deployed.

#### Configure



You can configure this feature using Helm. For information about configuring Network Policy, see Oracle Communications Cloud Native Core, Network Slice Selection Function Installation, Upgrade, and Fault Recovery Guide.

#### Observe

here is no specific metrics and alerts required for the Network Policy feature.

# 4.13 Validation of WWW-Authenticate Response Header 4xx with NSSF

When access token validation is enabled, NSSF performs access-token validation of the access token that comes with service requests to it. With this enhancement, NSSF has added supports 3GPP specified 4XX application error codes for these access token checks.

The access token validation include the following checks:

- Validating if access token is present in the service request: If the access token is not present, NSSF returns 401 unauthorized error code together with the "WWW-Authenticate" header as specified in 3GPP 16.5 29.531.
- 2. Validating if access token does not have the required scopes to invoke the service operation: NSSF validates the scope IE in AccessTokenClaims (which is the name of the NSSF services for which the access token is authorized) against the NSSF Service that are accessed in this service request. If the validation fails, NSSF returns a 403 Forbidden error code together with the "WWW-Authenticate" header as specified in 3GPP 16.5 29.531.

#### Managing Validation of WWW-Authenticate Response Header 4xx with NSSF

#### **Enable**

This feature does not require any configuration. It is enabled by default when the NSSF is installed.

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Network Slice Selection Function Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

# 4.14 Deleting Subscription on 404 SUBSCRIPITON\_NOT\_FOUND Response from AMF

This feature implements a mechanism for the Network Slice Selection Function (NSSF) to manage Access and Mobility Management Function (AMF) subscriptions related to Tracking Area (TA) status changes. When enabled, the NSSF attempts to notify the AMF of any status alterations within a specific TA. If the AMF returns a "404 Subscription Not Found" error, indicating the absence of an active subscription, the NSSF proceeds to delete the associated subscription. This ensures the NSSF maintains an accurate record of active subscriptions.



#### **Advantages:**

- Optimized Notification Flow: By removing inactive subscriptions, the system prevents the transmission of unnecessary notifications to AMFs that are not subscribed to specific TAs
- Resource Efficiency: Eliminating superfluous notifications reduces network traffic and processing overhead.
- Reduced Signaling Load: Prevents unnecessary signaling between the NSSF and AMF.

#### **Use Case Flow:**

- 1. TA Status Change: A status change occurs within a designated Tracking Area.
- 2. Notification Attempt: The NSSF attempts to notify the AMF of the status change.
- 3. AMF Response: The AMF responds to the notification attempt.
- 4. "404 Subscription Not Found" Response: If the AMF returns a "404 Subscription Not Found" error, it indicates there is no active subscription for the specific TA.
- Subscription Deletion: Upon receiving the "404" response, the NSSF deletes the subscription associated with the TA.
- Notification Cessation: The AMF no longer receives notifications regarding status changes for the deleted subscription's TA.



This behavior is applicable only when AMF responds with the '404 Subscription Not Found' error. It is not applicable to other failure scenarios.

## Managing Deleting Subscription on 404 SUBSCRIPITON\_NOT\_FOUND Response from AMF

#### **Enable**

To enable this feature, set the value of deleteOnSubscriptionNotFound parameter to true under the NSSubscription section in the ocnssf custom values 25.1.100.yaml file.

#### Observe

#### **Metrics**

The following metrics are used to provide information about this feature:

- ocnssf\_nssaiavailability\_notification\_delete\_on\_subscription\_not\_found\_total
- ocnssf\_nssaiavailability\_notification\_db\_error

For information about the Metrics, see NSSF Metrics.

#### **Error Scenarios**

The following error logs are generated for this feature:



Table 4-4 Error Scenarios

Microservice	Details
NsSubscription	Request URL:
	/nnssf-nssubscription/v1/nssai-availability/
	autoconfignotifications
	/nnssf-nssubscription/v1/nssai-availability/notifications
	Response Code/ Error Title:
	404 SUBSCRIPTION_NOT_FOUND
	Log Snippet:
	<pre>log snipper:  {     "instant": {         "epochSecond": 1661327119,         "nanoOfSecond": 10456886 },     "thread": "thread-1",     "level": "ERROR",     "loggerName": "com.oracle.cgbu.cne.nssf.nsselection.service .NsSubscriptionServiceImpl",     "message": "Failed to delete NssaiSubscribtion with ID: 1830762826",     "endOfBatch": false,     "loggerFqcn": "org.apache.logging.log4j.spi.AbstractLogger" ,     "contextMap": {},     "threadId": 54,     "threadPriority": 5,     "ts": "2022-08-24 07:45:19.010+0000",     "ooLogId": "\${ctx:ocLogId}",     "pod": "ocnssf-nssubscription-5f7bbbffbc-hxsfc",     "processId": "1",     "vendor": "Oracle",     "application": "ocnssf",     "engVersion": "22.3.0",     "mktgVersion": "22.3.0.0.0",     "microservice": "nssubscription",     "namespace": "ocnssf",     "node_name": "jazz-k8s-node-8"</pre>
	}



Table 4-4 (Cont.) Error Scenarios

Scenario	Microservice	Details
NsSubcription recieves 404	NsSubscription	Request URL:
SUBSCRIPTION_NOT_FOUND from client, Param		/nnssf-nssubscription/v1/nssai-availability/
deleteOnSubscriptionNotFound is		autoconfignotifications
true hence NssaiSubscription is		/nnssf-nssubscription/v1/nssai-availability/notifications
deleted		Response Code/ Error Title:
		404 SUBSCRIPTION_NOT_FOUND
		Log Snippet:
		{
		"instant": {
		"epochSecond": 1679654802,
		"nanoOfSecond": 234222276
		}, "thread": "task-3",
		"level": "ERROR",
		"loggerName":
		"com.oracle.cgbu.cne.nssf.nssubscription.serv
		ice.NsSubscriptionService",
		"message": "Recieved 404
		SUBSCRIPTION_NOT_FOUND from http://ocats-amf-
		<pre>stubserver.ocnssf:8080/notification/amf404/, subscriptionId = 402778803,</pre>
		deleteOnSubscriptionNotFound = true,
		response =
		NssfRestClientResponse{response=Response{prot
		ocol=h2_prior_knowledge, code=404, message=,
		url=http://ocnssf-egress-gateway:8080/
		OC_Notify/notification/amf404/},
		<pre>responseBody='{\"type\": \"NOT_FOUND\", \"title\": \"SUBSCRIPTION_NOT_FOUND\",</pre>
		\"status\": 404, \"detail\": \"subscription
		not found\", \"cause\":
		\"SUBSCRIPTION_NOT_FOUND\"}',
		messageCode=REMOTE_SERVER_EXCEPTION,
		<pre>eventTriggerSubMapId=0, attemptNum=0}",</pre>
		<pre>"endOfBatch": false,</pre>
		"loggerFqcn":
		"org.apache.logging.log4j.spi.AbstractLogger"
		<pre>"contextMap": {},</pre>
		"threadId": 760,
		"threadPriority": 5,
		"ts": "2023-03-24 10:46:42.234+0000",
		"ocLogId": "\${ctx:ocLogId}",
		"pod": "ocnssf-nssubscription-6d86c4d686-
		jbxjz",     "processId": "1",
		"vendor": "Oracle",
		"application": "ocnssf",
		"engVersion": "23.1.1-rc.2",



Table 4-4 (Cont.) Error Scenarios

Scenario	Microservice	Details
		<pre>"mktgVersion": "23.1.1-rc.2.0.0",     "microservice": "nssubscription",     "namespace": "ocnssf",     "node_name": "100.77.28.82"  } {     "instant": {         "epochSecond": 1679654802,         "nanoofSecond": 408253432     },     "thread": "XNIO-1 task-1",         "loggerName":     "com.oracle.cgbu.cne.nssf.nssubscription.help er.HelperFunctions",         "message": "Successfully deleted Subscription with ID: 402778803",         "endOfBatch": false,         "loggerFqcn":     "org.apache.logging.log4j.spi.AbstractLogger"     ,         "contextMap": {},         "threadId": 39,         "threadPriority": 5,         "ts": "2023-03-24 10:46:42.408+0000",         "ocLogId": "\${ctx:ocLogId}",         "pod": "ocnssf-nssubscription-6d86c4d686-jbxjz",         "processId": "1",         "vendor": "Oracle",         "application": "conssf",         "engVersion": "23.1.1-rc.2",         "mktgVersion": "23.1.1-rc.2.0.0",         "microservice": "nssubscription",         "namespace": "ocnssf",         "node_name": "100.77.28.82"</pre>
		} 



Table 4-4 (Cont.) Error Scenarios

Scenario	Microservice	Details
NsSubcription recieves 404	NsSubscription	Request URL:
SUBSCRIPTION_NOT_FOUND	'	/nnssf-nssubscription/v1/nssai-availability/
from client, Param deleteOnSubscriptionNotFound is		autoconfignotifications
false hence NssaiSubscription is		/nnssf-nssubscription/v1/nssai-availability/notifications
not deleted		Response Code/ Error Title:
		404 SUBSCRIPTION_NOT_FOUND
		Log Snippet:
		{
		instant": {
		"epochSecond": 1679655373,
		"nanoOfSecond": 880206537
		},
		"thread": "task-1",
		"level": "ERROR",
		"loggerName": "com.oracle.cgbu.cne.nssf.nssubscription.serv
		ice.NsSubscriptionService",
		"message": "Recieved 404
		SUBSCRIPTION_NOT_FOUND from http://ocats-amf-
		stubserver.devnssf-hrithik:8080/notification/
		amf404/, subscriptionId = 1871925794,
		deleteOnSubscriptionNotFound = false,
		<pre>response = NssfRestClientResponse{response=Response{prot</pre>
		ocol=h2_prior_knowledge, code=404, message=,
		url=http://ocnssf-egress-gateway:8080/
		OC_Notify/notification/amf404/},
		responseBody='{\"type\": \"NOT_FOUND\",
		\"title\": \"SUBSCRIPTION_NOT_FOUND\",
		<pre>\"status\": 404, \"detail\": \"subscription not found\", \"cause\":</pre>
		\"SUBSCRIPTION_NOT_FOUND\"}',
		messageCode=REMOTE_SERVER_EXCEPTION,
		eventTriggerSubMapId=0, attemptNum=0}",
		"endOfBatch": false,
		"loggerFqcn":
		"org.apache.logging.log4j.spi.AbstractLogger"
		,
		"threadId": 40,
		"threadPriority": 5,
		"ts": "2023-03-24 10:56:13.880+0000",
		"ocLogId": "\${ctx:ocLogId}",
		"pod": "ocnssf-nssubscription-9f68d8bc9-
		xsm22",     "processId": "1",
		"processid": "1",  "vendor": "Oracle",
		"application": "ocnssf",
		"engVersion": "23.1.1-rc.2",



Table 4-4 (Cont.) Error Scenarios

Scenario	Microservice	Details		
		<pre>"mktgVersion": "23.1.1-rc.2.0.0",     "microservice": "nssubscription",     "namespace": "ocnssf",     "node_name": "100.77.50.225"  } {     "instant": {         "epochSecond": 1679655373,         "nanoOfSecond": 908894025     },     "thread": "XNIO-1 task-1",     "level": "ERROR",     "loggerName": "com.oracle.cgbu.cne.nssf.nssubscription.service.NsSubscriptionService",         "message": "Not deleted NssaiSubscribtion with ID: 1871925794",         "endOfBatch": false,         "loggerFqcn": "org.apache.logging.log4j.spi.AbstractLogger",         "contextMap": {},         "threadId": 39,         "threadPriority": 5,         "ts": "2023-03-24 10:56:13.908+0000",         "ocLogId": "\${ctx:ocLogId}",         "pod": "ocnssf-nssubscription-9f68d8bc9-xsm22",         "processId": "1",         "vendor": "Oracle",         "application": "ocnssf",         "engVersion": "23.1.1-rc.2",         "mktgVersion": "23.1.1-rc.2.0.0",         "microservice": "nssubscription",         "namespace": "ocnssf",         "node_name": "100.77.50.225" }</pre>		

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- **1. Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Network Slice Selection Function Troubleshooting Guide.*
- 2. Raise a service request: See My Oracle Support for more information on how to raise a service request.



## 4.15 DNS SRV Based Selection of SCP in NSSF

NSSF selects Service Communication Proxy (SCP) for indirect communication of notifications using the static configurations done by the operator. This enhancement enables NSSF to learn SCP configuration from DNS SRV based FQDN, in addition to the already existing static manual configuration by the operator.

Egress Gateway (Egress Gateway) supports AlternateRoute Service, which NSSF is using to support DNS SRV based selection of SCP. It enables NSSF to resolve FQDN or Virtual FQDN to alternate FQDNs of SCP. Egress Gateway uses the virtual FQDN of SCP instances to guery the AlternateRoute Service and get the list of alternate FQDNs with priorities assigned to each of them. Based on the priorities, Egress Gateway picks up the SCP instances for rerouting attempts.

The AlternateRoute Service allows the configuration of multiple sets of SCP instances in NSSF in contrast to only one static configuration in the previous scenario.

#### Managing DNS SRV Based Selection of SCP in NSSF

#### Configure

#### **Configure Using Helm Parameters:**

DNS SRV is enabled by default at the time of installation. The dnsSrvEnabled parameter is set to true by default in the ocnssf\_custom\_values\_25.1.100.yaml file:

dnsSrvEnabled: true



#### Note

The default value of this parameter is set to true by default. It is recommended to keep this value as true only. Disabling it may cause issues with the functioning of this feature and other features that are depended on it.

dnsSrvFqdnSetting.enabled: true



#### (i) Note

Flag to enable or disable the usage of custom patterns for the FQDN while triggering DNS-SRV query. It is set to true by default.

dnsSrvFqdnSetting.pattern: "\_{scheme}.\_tcp.{fqdn}."

For more information about Helm parameters to configure DNS SRV and Alternate Routing Service, see "Alternate Route Microservice Parameters section" in Oracle Communications Cloud Native Core, Network Slice Selection Function Installation, Upgrade, and Fault Recovery Guide.

#### **Configure Using REST API:**

The feature related REST API configurations are performed at NSSF and Egress Gateway.

Perform REST API configurations at Egress Gateway in the following sequence:



Configure peerconfiguration to define the list of peers to which Egress Gateway can send request.

#### (i) Note

It is mandatory to configure peerconfiguration with healthApiPath if you want to enable peermonitoringconfiguration.

- Configure virtualHost under peerconfiguration where the AMF query is sent.
- Configure peersetconfiguration to logically group the peers into sets.

#### (i) Note

- peerIdentifier must be the value of SCP peer configured in peerConfiguration.
- You cannot configure multiple virtual hosts as peers in the same peer set.
- Configure the priority for each SCP peer in the set. Depending on the priority, it selects the primary, secondary, or tertiary SCP peers to route requests.
- d. Configure or update errorcriteriasets.
- e. Configure or update erroractionsets.
- Configure routesconfiguration to define the route and reroute parameters. If SBIRouting functionality is required, then configure SBIRoutingFilter. If reroute mechanism is required for that route, then configure SBIReroute filter with retries, methods, and statuses.



#### (i) Note

peerSetIdentifier must be the value configured during peersetconfiguration.

Set the value of enabled under sbiRoutingConfiguration to true to route the AMF queries through SCP configured in the id attribute.



#### (i) Note

peerconfiguration and peersetconfiguration can be either set to empty list or populated with values. These attributes are used for routing only if sbiRoutingConfiguration is enabled for a particular route.

h. <Optional> You can also configure peermonitoringconfiguration using REST API or CNC Console. For more information about enabling or configuring peermonitoringconfiguration, see Monitoring the Availability of SCPs using SCP Health APIs.





#### (i) Note

It is mandatory to configure peerconfiguration with healthApiPath if peermonitoringconfiguration is enabled.

- 2. Perform the following REST API configurations at NSSF:
  - Configure the nssaiauth Managed Object to enable the configuration of network slice authentication rules by configuring Grant status (Allowed PLMN, Rejected PLMN, or Rejected TAC) for S-NSSAI on a per TAI basis.

For more information about REST API parameters and configuration, see Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.

#### **Configure using CNC Console**

Taking into account the prerequisites criteria and recommended sequence explained in sections above, you can see Configuring NSSF using CNC Console section for configuring this feature using CNC Console.

#### **Observe**

#### Metrics

No new Metrics or KPIs were added to NSSF. However, the following Egress Gateway metrics for Alternate Route Service are used to provide the information about this feature:

- oc fqdn alternate route total
- oc\_dns\_srv\_lookup\_total
- oc\_alternate\_route\_resultset
- oc\_configclient\_request\_total
- oc configclient response total

For information about the Metrics, see Egress Gateway Metrics in NSSF Metrics.

#### **Error Scenarios**

No new logs are generated for this feature. However, it uses the following Egress Gateway error scenarios:



Table 4-5 Error Scenarios

Scenario	Microservice	Details	
Sending Subscription notification	ocnssf-egress-gateway Request URL:		
failed due to UnknownHost		/nnssf-configuration/v1/nssaiauth	
exception		Response Code/ Error Title:	
		503	
		Service Unavailable	
		Encountered unknown host exception at Egress Gateway	
		Log Snippet:	
		{	
		"instant": {	
		"epochSecond": 1676964069,	
		"nanoOfSecond": 986866249	
		},	
		"thread": "@6c8fe7a4-217",	
		"level": "ERROR",	
		"loggerName":	
		"ocpm.cne.gateway.jettyclient.DnsResolver",	
		"message": "UnExpected error occured ::	
		{}",	
		<pre>"thrown": {     "commonElementCount": 0,</pre>	
		"localizedMessage": "ocats-amf-	
		stubserver.ocnssf: Name or service not	
		known",	
		"message": "ocats-amf-	
		stubserver.ocnssf:Name or service not known",	
		"name": "java.net.UnknownHostException",	
		<pre>"extendedStackTrace": [</pre>	
		{	
		"class": "java.net.Inet6AddressImpl",	
		"method": "lookupAllHostAddr",	
		"file": "Inet6AddressImpl.java",	
		"line": -2,	
		<pre>"exact": false,     "location": "?",</pre>	
		"version": "?"	
		},	
		\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	
		"class":	
		"java.net.InetAddress\$PlatformNameService",	
		"method": "lookupAllHostAddr",	
		"file": "InetAddress.java",	
		"lin e": 933,	
		"exact": false,	
		"location": "?",	
		"version": "?"	
		<b> </b>	
		<b>l</b> 1	
		] },	



Table 4-5 (Cont.) Error Scenarios

Scenario	Microservice	Details		
		<pre>"endOfBatch": false,   "loggerFqcn": "org.apache.logging.log4j.spi.AbstractLogger" ,   "contextMap": {},   "threadId": 217,   "thread Priority": 5,   "messageTimestamp": "2023-02-21T07:21:09.986+0000",   "ocLogId": "\${ctx:ocLogId}",   "pod": "\${ctx:hostname}",   "processId": "1",   "instanceType": "prod",   "egressTxId": "\${ctx:egressTxId}" }</pre>		

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- **1. Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

## 4.16 OAuth Access Token Based Authorization

NSSF supports Oauth 2.0, which is a security feature that NSSF uses to validate and authorize requests from allowed or valid consumers NFs. The consumer NF requests for access token from the issuer NRF, and uses this access token to send the request to NSSF. NSSF validates the requests and approves or discards it based on access token authorization received in the request. The access token is validated with the configured public key certificate in NSSF.

Before this enhancement, NSSF used NRF Instance ID to validate the access token, where Ingress Gateway stored public keys against NRF instance Id. This enhancement allows NSSF to use multiple public certificates for validating access tokens by adding support for Key-ID (K-ID) based access token validation, in addition to the existing NRF Instance ID based access token validation.

This enhancement now allows Ingress Gateway to operate in the following three different modes:

- K-ID based ONLY
  - Ingress Gateway validates access token based on public keys indexed with key-id only.
- 2. Instance ID based ONLY (DEFAULT)



- Ingress Gateway validates access token based on public keys indexed with NRF Instance ID in the issuer field.
- K-ID based with Instance ID based as fallback (KID\_PREFERRED)
  - a. Ingress Gateway validates access token based on public keys indexed with Key-ID. If Key-ID is not FOUND in Access token, Ingress Gateway attempts token validation using public keys indexed with NRF instance ID in the issuer field.
  - b. Fallback happens only if the received access token is structured as follows:
    - i. Does not contain Key-ID
    - ii. Contains Key-ID but does not have public keys configured against the Key-ID

#### Managing OAuth Access Token Based Authorization Using Key-ID and NRF Instance ID

#### **Prerequisites**

This section describes the configurations required to enable access tokens before deploying NSSF.

#### Generating KeyPairs for NRF Instances



It is at the discretion of the user to create private keys and certificates, and it is not in the scope of NSSF. This section lists only samples to create KeyPairs.

Using the OpenSSL tool, the user can generate private key and public certificates. The commands to generate the KeyPairs are as follows:

#### Example Command to generate KeyPair for NRF Instance

```
openssl ecparam -genkey -name prime256v1 -noout -out ec_private_key1.pem

openssl pkcs8 -topk8 -in ec_private_key1.pem -inform pem -out
ec_private_key_pkcs8.pem -outform pem -nocrypt

openssl req -new -key ec_private_key_pkcs8.pem -x509 -nodes -days 365 -out
4bc0c762-0212-416a-bd94-b7f1fb348bd4.crt -subj "/C=IN/ST=KA/L=BLR/O=ORACLE/OU=CGBU/CN=ocnrf-endpoint.ocnrf.svc.cluster.local"
```

#### Note

For ATS configuration details, see **Configuring Secrets to Enable Access Token** in **Preinstallation Tasks** of *Cloud Native Core Network Slice Selection Function Installation and Upgrade* Guide.

#### **Enabling and Configuring Access Token**

To enable access token validation, configure both Helm-based and REST-based configurations on Ingress Gateway.





While Helm based configuration is mandatory, you can also perform CNC Console-based configuration instead of REST-based configurations.

#### **Configuration using Helm:**

For Helm-based configuration, perform the following steps:

1. Create a secret that stores NRF public key certificates using the following commands:

kubectl create secret generic <secret name> --from-file=<filename.crt> -n
<Namespace>

#### For Example:

kubectl create secret generic oauthsecret --from-file=4bc0c762-0212-416abd94-b7f1fb348bd4.crt -n ocnssf

#### Note

In the above command:

- oauthsecret is the secret name
- ocnssf is the namespace
- 4bc0c762-0212-416a-bd94-b7f1fb348bd4.crt is the public key certificate
- 2. Enable the <code>oauthValidatorEnabled</code> parameter on Ingress Gateway by setting its value to true. Further, configure the secret and namespace on Ingress Gateway in the OAUTH CONFIGURATION section of the <code>ocnssf\_custom\_values\_25.1.100.yaml</code> file using the following fields:
  - oauthValidatorEnabled
  - nfType
  - nfInstanceId
  - producerScope
  - allowedClockSkewSeconds
  - enableInstanceIdConfigHook
  - nrfPublicKeyKubeSecret
  - nrfPublicKeyKubeNamespace
  - validationType
  - producerPlmnMNC
  - producerPlmnMCC
  - oauthErrorConfigForValidationFailure
  - oauthErrorConfigForValidationFailure.errorCode



- oauthErrorConfigForValidationFailure.errorTitle
- oauthErrorConfigForValidationFailure.errorDescription
- oauthErrorConfigForValidationFailure.errorCause
- oauthErrorConfigForValidationFailure.redirectUrl
- oauthErrorConfigForValidationFailure.retryAfter
- oauthErrorConfigForValidationFailure.errorTrigger
- oauthErrorConfigForValidationFailure.errorTrigger.exceptionType

#### Note

4bc0c762-0212-416a-bd94-b7f1fb348bd4.crt is the public key certificate and we can have any number of certificates in the secret.

The following snippet represents the location of the mentioned parameter in the Helm file:

#### ① Note

- The following snippet represents only the sample values.
- For more information on parameters and their supported values, see Ingress
  Gateway Parameters from Customizing NSSF chapter in Oracle
  Communications Cloud Native Core, Network Slice Selection Function
  Installation, Upgrade, and Fault Recovery Guide
- For information about OAuth access token attributes like kid, typ, iss, aud, scope etc., see <a href="https://www.rfc-editor.org/rfc/rfc7515.html">https://www.rfc-editor.org/rfc/rfc7515.html</a> page.

```
#OAUTH CONFIGURATION
```

producerPlmnMNC: 14

```
oauthValidatorEnabled: true

nfType: NSSF

nfInstanceId: 9faf1bbc-6e4a-4454-a507-aef01a101a01

producerScope: nnssf-nsselection,nnssf-nssaiavailability

allowedClockSkewSeconds: 0

enableInstanceIdConfigHook: true

nrfPublicKeyKubeSecret: oauthsecret

nrfPublicKeyKubeNamespace: ocnssf

validationType: strict
```



```
producerPlmnMCC: 310
  oauthErrorConfigForValidationFailure:
   errorCode: 401
   errorTitle: "Validation failure"
   errorDescription: "UNAUTHORIZED"
    errorCause: "oAuth access Token validation failed"
   redirectUrl:
   retryAfter:
   errorTrigger:
    - exceptionType: OAUTH_CERT_EXPIRED
      errorCode: 408
      errorCause: certificate has expired
      errorTitle:
      errorDescription:
      retryAfter:
      redirectUrl:- exceptionType: OAUTH_MISMATCH_IN_KID
      errorCode: 407
      errorCause: kid configured does not match with the one present in
the token
      errorTitle:
      errorDescription:
      retryAfter:
      redirectUrl:
    - exceptionType: OAUTH_PRODUCER_SCOPE_NOT_PRESENT
      errorCode: 406
      errorCause: producer scope is not present in token
      errorTitle:
      errorDescription:
      retryAfter:
```



```
redirectUrl:
    - exceptionType: OAUTH_PRODUCER_SCOPE_MISMATCH
      errorCode: 405
      errorCause: produce scope in token does not match with the
configuration
      errorTitle:
      errorDescription:
      retryAfter:
      redirectUrl:
    - exceptionType: OAUTH_MISMATCH_IN_NRF_INSTANCEID
      errorCode: 404
      errorCause: nrf id configured does not match with the one present in
the token
      errorTitle:
      errorDescription:
      retryAfter:
      redirectUrl: - exceptionType: OAUTH_PRODUCER_PLMNID_MISMATCH
      errorCode: 403
      errorCause: producer plmn id in token does not match with the
configuration
      errorTitle:
      errorDescription:
      retryAfter:
      redirectUrl:
    - exceptionType: OAUTH_AUDIENCE_NOT_PRESENT_OR_INVALID
      errorCode: 402
      errorCause: audience in token does not match with the configuration
      errorTitle:
      errorDescription:
```



```
retryAfter:
    redirectUrl:
    - exceptionType: OAUTH_TOKEN_INVALID
    errorCode: 401
    errorCause: oauth token is corrupted
    errorTitle:
    errorDescription:
    retryAfter:
redirectUrl:oauthErrorConfigOnTokenAbsence:
    errorCode: 400
    errorTitle: "Token not present"
    errorDescription: "UNAUTHORIZED"
    errorCause: "oAuth access Token is not present"
    redirectUrl:
    retryAfter:
```

#### Configuration using REST API or CNC Console

#### **REST API**

After Helm configuration, send the REST requests to use configured public key certificates. Using REST-based configuration, you can distinguish between the certificates configured on different NRFs and can use these certificates to validate the token received from a specific NRF.

For more information about REST API configuration, see **OAuth Validator Configuration** section in *Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.* 

#### **CNC Console**

For more information on CNC Console based configuration, see <a href="OAuth Validator">OAuth Validator</a> Configurations.

#### **Observe**

- Added the following success measurements:
  - oc\_oauth\_nrf\_request\_total
  - oc\_oauth\_nrf\_response\_success\_total
  - oc\_oauth\_token\_cache\_total
  - oc\_oauth\_validation\_successful\_total



- oc\_oauth\_cert\_expiryStatus
- oc\_oauth\_cert\_loadStatus
- oc.oauth.keyid.count
- Added the following error measurements:
  - oc\_oauth\_nrf\_response\_failure\_total
  - oc\_oauth\_request\_failed\_internal\_total
  - oc\_oauth\_request\_invalid\_total
  - oc\_oauth\_validation\_failure\_total
  - oc.oauth.request.failed.cert.expiry

For information on Metrics and KPIs of NSSF, see <u>NSSF Metrics</u> and <u>NSSF KPIs</u> sections respectively.

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

## 4.17 Overload Control based on Percentage Discards

The Overload Control feature protects the system from overload and maintains the overall health of NSSF. The system needs to not only detect overload conditions but also protect against the same. Further, it needs to mitigate against and avoid the system from entering into an overload condition by taking necessary actions for recovering from overload.

NSSF provides the following means for overload management:

- Predefined threshold load levels
- Tracks number of pending messages
- Tracks CPU and memory usage
- Enforce load shedding during various overload levels

Perf-info performs overload calculations based on the indicators:

- CPU Utilization
- Memory Utilization
- Pending Message Count
- Failure Count

The overload level is configured for the following NSSF microservices:

- NSSelection
- NSAvailability

The Overload Manager module in Perf-info is configured or updated with the threshold value for services. A configurable flag is available for sampling interval as



ocPolicyMapping.samplingPeriod based on which Ingress Gateway calculates rate per service in the current sampling period and applies appropriate discard policies and actions in the subsequent sampling period.

Overload Manager triggers Rate Calculator to start calculating the rate of incoming requests per service per sampling period. Ingress Gateway receives a notification event per service with the calculated rates to the Overload Manager filter at the end of every sampling period. It applies an appropriate configured discard policy for a particular service based on the rate of requests.

Ingress Gateway calculates the number of requests to be dropped in the current sampling period based on configured percentage discard.

Overload Thresholds for each service is evaluated based on four metrics namely  $\mathtt{cpu}$ ,  $\mathtt{svc\_failure\_count}$ ,  $\mathtt{svc\_pending\_count}$ , and  $\mathtt{memory}$ . Overload control is triggered if the thresholds for any one metrics are reached.

#### Note

When the percentage-based overload control discarding policy is enabled, the number of requests to be dropped in the current sampling period is computed based on the configured percentage discard and the "rate of requests outgoing of Ingress Gateway" in the previous sampling period for the service.

Once the number of requests to be dropped in the current sampling period is computed, the gateway does not drop all the new traffic to meet the discard count. Instead, Ingress Gateway executes a random function to decide if a request is to be discarded or not. If the random function returns true, the request is discarded in the current sampling period with the discard action "RejectWithErrorCode". This ensures there is a spread of discard requests in a sampling period.

Since we are calculating the number of requests to be dropped in the current sampling period based on the number of requests sent to the backend service in the previous sampling period and not on the total requests received at Ingress Gateway, the percentage dropped is not exactly the percentage configured.

#### **Managing Overload Control based on Percentage Discards**

#### **Enable**

You can enable Overload Control feature using the following Helm configuration:

- 1. Open the ocnssf\_custom\_values\_25.1.100.yaml file.
- 2. Set the global.performanceServiceEnable parameter to true in the ocnssf\_custom\_values\_25.1.100.yaml file.

The following snippet represents the location of the mentioned parameter in the Helm file:

#Flag to Enable or Disable Performance Service. The flag is set to true to
enable the overload control feature by default.
 performanceServiceEnable: true

3. Set the perf-info.overloadManager.enabled parameter to true in the ocnssf custom values 25.1.100.yaml file.



The following snippet represents the location of the mentioned parameter in the Helm file:

```
overloadManager:
   ingressGatewayPort: *httpSignalPort
   #Flag to Enable or Disable overloadManager
   enabled: true
```

**4.** Configure the Prometheus URI in perf-info.configmapPerformance.prometheus The following snippet represents the location of the mentioned parameter in the Helm file:

```
perf-info
  configmapPerformance:
    prometheus: http://occne-prometheus-server.occne-infra:80
```



Update the URL as per your setup. It should be a valid Prometheus server URL, which is same as data source URL used on the Grafana dashboard.

- 5. Save the ocnssf\_custom\_values\_25.1.100.yaml file.
- 6. Run helm upgrade, if you are enabling this feature after NSSF deployment. For more information on upgrade procedure, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Installation and Upgrade Guide*.

#### Configure

You can configure this feature using Helm, REST API, and CNC Console.

#### **Configure using REST API:**

The Overload Control feature related configurations are performed at Ingress Gateway and Perf-info.

The following REST APIs must be configured for this feature in the following order:

{apiRoot}/nssf/nf-common-component/v1/igw/errorcodeprofiles



**Dependency**: errorcodeprofiles is used in ocdiscardpolicies to define how different overload levels trigger rejection with specific errors.

{apiRoot}/nssf/nf-common-component/v1/igw/ocdiscardpolicies

#### (i) Note

#### Dependnecy:

- ocdiscardpolicies use errorcodeprofiles to decide the error message when rejecting requests.
- ocdiscardpolicies are used by ocpolicymapping to associate services with specific overload handling policies.



{apiRoot}/nssf/nf-common-component/v1/igw/ocpolicymapping

(i) Note

## Dependnecy:

- Depends on ocdiscardpolicies to apply the right rejection policy to each service.
- {apiRoot}/nssf/nf-common-component/v1/igw/errorcodeserieslist



#### (i) Note

Not directly linked to other configurations but enhances error handling by classifying errors.

{apiRoot}/nssf/nf-common-component/v1/igw/routesconfiguration



#### (i) Note

routesconfiguration defines routing behaviors and associates services with error code series. It references errorCodeSeriesId which is defined in id attribute in errorcodeserieslist.

{apiRoot}/nssf/nf-common-component/v1/perfinfo/overloadLevelThreshold



#### Note

Determines when a service is considered overloaded. It triggers ocdiscardpolicies when thresholds are exceeded, causing requests to be rejected with predefined error responses.

For more information about APIs, see Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide

#### **Configure using CNC Console:**

The above REST APIs can also be configured using CNC Console UI. For more information, see the follwing sections in the Configuring NSSF using CNC Console chapter:

- Create or update the Error Code Profiles.
- Create or update the Create Overload Control Discard Policies.
- Create or update the Discard Policy Mapping.
- Create or update the Error Code Series.
- Update the Routes Configuration.

#### **Observe**

#### Metrics

No new metrics added to NSSF for the Overload Control feature. However, the following Perfinfo metrics are used to provide the information about overload control feature:



- cgroup\_cpu\_nanoseconds
- cgroup\_memory\_bytes
- load\_level

For information about Metrics, see Perf-info metrics for Overload Control in NSSF Metrics.

For information on KPIs of NSSF, see NSSF KPIs section.

#### **Alerts**

The following alerts are added for the Overload Control feature:

- OcnssfOverloadThresholdBreachedL1
- OcnssfOverloadThresholdBreachedL2
- OcnssfOverloadThresholdBreachedL3
- OcnssfOverloadThresholdBreachedL4

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

## 4.18 Autopopulation of Configuration Using NRF Content

NSSF responds with CandidateAMFList (list of possible AMFs ) to Initial Registration request. Currently, the computation of active AMFs in AMF Set is done by one of the following methods:

- Discovery: For each registration NSSF sends a discovery message to NRF to get all active AMFs in a AMF-Set.
- Operator configuration: The operator has to ensure that all active AMFs are mapped to the AMF Set.

NSSF must maintain the information about AMF Set to AMF mapping in the database as it uses this information while responding to NSSelection initial registration query.

Before this enhancement, the operator was required to maintain this information manually in the database. This enhancement removes the need for the manual configuration by allowing NSSF to autoconfigure AMF information using NRF (Network Repository Function) whenever there is an update in the AMF Set data. NSSF automatically determines AMF information in a AMF-Set and autopopulate NSSF configuration using the information from NRF.

- For each initial registration, NSSF sends a discovery message to NRF to get AMFs in a AMF-Set.
- 2. NSSF maintains AMF-Set (MCC-MNC-SetId-RegionId) to AMF list (List of AMFs, which belong to a AMF-Set in NSSF DB) mapping.
  - a. Subscription based on AMF-Set: NSSF sends a discovery and subscribe request to NRF based ontheAMF-Set configured by the operator and maintain AMF-Set to AMFs mapping in NSSF Database.



Hence, operator is now required to configure only AMF set. The information about active amfs are maintained by NSSF, as it is autoconfigured using NRF content. This resolves the issue of stale configuration and NSSF does not have to send discovery for each initial registration request, saving CPU and network bandwidth.

#### **Managing Autopopulation of Configuration Using NRF Content**

#### **Enable**

This section provides the procedure to enable this feature:

1. Set the nsconfig.nrf.subscription parameter to true in the ocnssf\_custom\_values\_25.1.100.yaml file.

The following snippet represents the location of the mentioned parameter in the Helm file:

```
nsconfig:
    nrf:
    subscription: false # Flag to enable Subscriptions towards NRF for
AmfSet
```

2. Set the nsselection.features.candidateResolution parameter to true in the ocnssf custom values 25.1.100.yaml file.

The following snippet represents the location of the mentioned parameter in the Helm file:

```
nsselection:
    features:
    candidateResolution : false #Flag to true and false to enable or
disable Candidate Resolution feature
```

#### (i) Note

- When this feature is set to false, NSSF returns TargetAMFSetId and TargetAMFRegionId for NSSelection GET request for Initial Register message and UE-Config update.
- When this feature is set to true, NSSF computes and returns Candidate AMF list for NSSelection GET request for Initial Register message and UE-Config update.

#### Configure

#### **Configure using Helm Parameters:**

No additional helm configuration is required to enable this feature.

#### **Configure using REST API:**

There is no option to enable or disable this feature using REST API configuration.

However, this feature will use existing AMF set and NSI-Profile REST APIs configurations if nsconfig.nrf.subscription is set to true in the ocnssf\_custom\_values\_25.1.100.yaml file.



#### Note

- When nsconfig.nrf.subscription is set to true ocnssf\_custom\_values\_25.1.100.yaml file, even if corresponding AMF set is not configured for an NSI-Profile, this feature will work.
- For more information on REST APIs, see *Oracle Communications Cloud Native Core*, *REST API Guide*.

#### Configure using CNC Console:

There are no CNC Console configurations to enable this feature. However, this feature will use existing AMF set and NSI-Profile configurations, which can be done using REST API or CNC Console. For more information about CNC Console configuration, see <a href="AMF Set">AMF Set</a> and <a href="AMF Set">NSI</a> Profile.

#### **Observe**

#### Metrics

Added the following success measurements:

- ocnssf\_nsconfig\_nrf\_disc\_success\_total
- ocnssf\_subscription\_nrf\_tx\_total

#### Added the following error measurements:

- ocnssf\_nsconfig\_nrf\_disc\_error\_total
- ocnssf\_discovery\_nrf\_tx\_failed\_total
- ocnssf\_subscription\_nrf\_tx\_failed\_total

For information on Metrics and KPIs of NSSF, see <u>NSSF Metrics</u> and <u>NSSF KPIs</u> sections respectively.

#### **Alerts**

There are no new alerts for this feature.

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. Collect the logs: For more information on how to collect logs, see Oracle Communications Cloud Native Core, Network Slice Selection Function Troubleshooting Guide.
- **2.** Raise a service request: See My Oracle Support for more information on how to raise a service request.

# 4.19 Auto-Population of Configuration Based on NSAvailability Update

The Auto-Population of Configuration Based on NSAvailability Update feature allows the Network Slice Selection Function (NSSF) to automatically learn and adapt its configuration for supported S-NSSAIs (Network Slice Selection Service Area Identifiers) from trusted Access



and Mobility Functions (AMFs). This automation minimizes manual configuration efforts for operators by filling in the gaps where the operator has not specified allowed S-NSSAIs.

#### **Benefits**

- Reduces Manual Configuration: Focuses operators' efforts on restricted S-NSSAIs.
- Minimizes Errors: Reduces potential errors from manual configuration.
- Keeps Configurations Up-to-Date: Ensures NSSF configurations align with trusted AMF capabilities.

#### **How It Works?**

- AMF Sends Availability Update: An AMF sends a message to NSSF about the S-NSSAIs
  it supports in a specific Tracking Area Identifier (TAI).
- 2. NSSF Checks Configuration: NSSF checks its provisional database to determine if the S-NSSAIs are already configured for that TAI.
- Automatic Configuration (if enabled and applicable):
  - If enabled and the S-NSSAIs are not configured for that TAI:
    - NSSF creates entries in the nssai\_auth table for the S-NSSAIs and TAI.
    - NSSF creates an nss\_rule for the S-NSSAIs with access type set to 3GPP (if supported in one or more TAIs).
    - NSSF notifies other AMFs subscribed to the TAI about the new S-NSSAIs.
  - If the S-NSSAIs are already configured or if the feature is disabled, NSSF takes no action on the AMF's update.

## Sample Call Flow Illustration Scenario

- The operator configures NSSF to allow S-NSSAI "S1" in TAI "1".
- A trusted AMF sends an NSAvailability Update indicating it supports S-NSSAIs "S1" and "S2" in TAI "1".

#### **Call Flow (With This Feature Disabled)**

- 1. **AMF to NSSF**: AMF sends the NSAvailability Update.
- 2. NSSF:
  - a. Stores the update in StateDB (dynamic configuration).
  - b. Checks ProvisionDB (operator configuration) and finds only "S1" is allowed in TAI "1".
  - c. Validates and authorizes "S1" based on ProvisionDB.
- 3. NSSF to AMF: NSSF responds with "SNSSAI S1 is authorized in TAI-1".

### Call Flow (With This Feature Enabled)

- 1. **AMF to NSSF**: AMF sends the NSAvailability Update.
- 2. NSSF:
  - Stores the update in StateDB.
  - b. Checks ProvisionDB and finds only "S1" is allowed in TAI "1".
  - c. Identifies "S2" as not configured in ProvisionDB for TAI "1".
  - d. Creates an entry in the nssai\_auth table for TAI "1" and S-NSSAI "S2" with grant set to "ALLOWED".



- e. Creates an nss\_rule associating the nssai\_auth entry with the PLMN level profile of TAI "1" (for 3GPP access).
- f. Sends notification to other AMFs subscribed to TAI "1" about the new S-NSSAI "S2".
- NSSF to AMF: NSSF responds with "SNSSAI S1, S2 is authorized in TAI-1".

#### **Key Differences**

- With Feature Enabled: NSSF learns about "S2" from the AMF and automatically configures it for TAI "1".
- With Feature Disabled: NSSF only authorizes "S1" based on preconfigured operator settings.

#### **Additional Tasks**

 Delete S-NSSAI Entry: When an AMF sends a delete request, NSSF checks if any other AMF in the region supports the S-NSSAI. If not, NSSF deletes the corresponding nssai\_auth and all nss\_rule entries.

#### Managing Auto-Population of Configuration Based on NSAvailability Update

#### **Enable**

If the prerequisites are fulfilled, this feature can be enabled or disabled using **NSSF System Options** REST API by setting AutoConfigurationFromNsAvailability parameter to one of the following possible values:

- Disable: This signifies the feature is disabled and is the default value. There will be no configuration update based on NsAvailability data if the value is disabled.
- FromTrustedAMFs: This signifies NsAvailability Update only from trusted AMFs may lead to an update in NSSF configuration.
- FromAllamFs: This signifies that NsAvailability Update from any AMF may lead to an update in NSSF configuration.

#### Note

Assuming that NsAvailability Update is processed by NSSF. That is, the AMF is authorized to send the availability update.

#### Steps to Enable

- Get NSSF System Options: Retrieve the current system options.
- Update NSSF System Options: Change the value of
   AutoConfigurationFromNsAvailability to FromTrustedAMFs or FromAllaMFs based on
   the requirement.
- Put NSSF System Options: Send an HTTP PUT request with the updated system options.



#### Note

- Significance of NSSF System Options: This managed object enables the feature only when AutoConfigurationFromNsAvailability is set to FromTrustedAMFs or FromAllAMFs. Based on the set value, NSSF configuration updates based on NSAvailabilityUpdate will either occur from trusted AMFs only or from all AMFs.
- For more information, see **NSSF System Options** in *Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.*
- For CNC Console-based configuration, see <u>NSSF System Option</u> section.

#### **Prerequisites**

Operator must configure the following for this feature to work:

Configure PLMN Level NSI Profile: Configure PLMN Level NSI Profile for each supported PLMN, as nssai\_auth autoconfiguration happens only when default profile is configured for the PLMN. For more information on REST based configuration, see PLMN Level NSI Profile in Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide. For more information on CNC Console based configuration, see PLMN Level NSI Profile in 'Configuring NSSF using CNC Console'.

#### ① Note

- Significance of PLMN Level Profile: If the PLMN level NSI profile is not configured, NSSF cannot create configuration based on NSAvailability Update as the message does not provide slice instance details. NSSF responds with an error if this configuration is missed and a configuration update is required.
- For more information, see PLMN Level NSI Profile in Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.
- For CNC Console-based configuration, see PLMN Level NSI Profiles.

#### 2. Configure AMF Set:

Configure the **AMF Set** under which the **Trusted AMF** falls as a prerequisite before creating or updating a Trusted AMF.

#### Note

- For more information, see **AMF Set** in *Oracle Communications Cloud Native Core*, *Network Slice Selection Function REST Specification Guide*.
- For CNC Console-based configuration, see <u>AMF Set</u>.
- 3. Configure Trusted AMFs:





#### (i) Note

This managed object is considered only when the value of AutoConfigurationFromNsAvailability parameter is set to FromTrustedAMFs in NSSF System Options REST API.

This managed object is used to configure a list of AMFs from which the configuration is trusted. If these AMFs support one or more S-NSSAIs in TAI/TAIs, NSSF updates its own configuration and signifies those S-NSSAIs are supported in those TAIs.

#### (i) Note

- NSSF accepts and provisions the configuration only when the request is from trusted AMFs. If the Managed Object is not configured, or if all trusted AMFs are not provided as input, the NSAvailability Update from those AMFs will not impact NSSF's provisional configuration. NSSF will not update the provision database based on NSAvailability Update from any AMF and will behave according to the spec without updating the provision configuration.
- NSSF treats the Availability PUT as equivalent to operator configuration, updating its configuration and notifying other AMFs of new S-NSSAIs.
- For more information, see Trusted AMF in Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.
- For CNC Console-based configuration, see <u>Trusted Amf</u>.
- 4. Configure **Nss Rule** for SNSSAIs supported for Non 3GPP accessType. This enhancement configures nss rules for 3GPP AccesType only. Rules for SNSSAIs supported for Non 3GPP accessType must be still configured by Operator.

#### (i) Note

- For more information, see Nss Rule in Oracle Communications Cloud Native Core. Network Slice Selection Function REST Specification Guide.
- For CNC Console-based configuration of NSS Rule, see "Configuring NSSF using CNC Console" chapter in Oracle Communications Cloud Native Core, Network Slice Selection Function User Guide.

#### **⚠** Caution

- **GR Deployment**: In a GR deployment, the operator must configure the same set of AMFs on all sites.
- **Security:** Trusted AMFs can update NSSF configuration, so operators must carefully manage which AMFs are given this capability.
- Site-Level Configuration: Trusted AMFs and system options are site-level configurations. Operators must ensure consistent configuration across all sites.
- Non-3GPP Access Type: This enhancement configures Nss Rule for 3GPP AccessType only. Operators must configure **Nss Rule** for Non-3GPP AccessType.



#### Observe

No new metrics or KPIs are generated for this feature. For information on other Metrics and KPIs of OCNSSF, see OCNSSF Metrics and OCNSSF KPIs sections respectively.

#### **Error Scenarios**

For Auto-Population of Configuration Based on NSAvailability Update, logs are generated for NSAvailability, when error is due to configuration in the PLMN Level NSI Profile.



Table 4-6 Error Scenarios

Scenario	Microservice	Details
PLMN Level NSI Profile is not	Nnssf_NSSAIAvailabilit	Response Code/ Error Title:
configured	у	Configuration issue: PLMN Level Profile is not configured for <mcc> <mnc></mnc></mcc>
		Unable to process nsavailability request
		500 Response with details missing configuration.
		Unable to find PLMN level profile for <mcc> <mnc></mnc></mcc>
		Log Snippet:
		<pre>"instant": {     "epochSecond": 1661325081,     "nanoofSecond": 495990110 },     "thread": "XNIO-1 task-1",     "level": "ERROR",     "loggerName": "com.oracle.cgbu.cne.nssf.nsavailability.data servicehelper.AmfTaiSnssaiMapDataPopulation",     "message": "CONFIGURATION_ERROR: Unable to find Plmn Level Profile",     "endOfBatch": false,     "loggerFqcn": "org.apache.logging.log4j.spi.AbstractLogger" ,     "contextMap": {         "ocLogId": "1661325081404_2998_ocnssf- ingress-gateway-fd65885d6-lppss" },     "threadId": 234,     "threadPriority": 5,     "ts": "2022-08-24 07:11:21.495+0000",     "ocLogId": "1661325081404_2998_ocnssf- ingress-gateway-fd65885d6-lppss",     "pod": "ocnssf-nsavailability-65466b5f48- xtvgz",     "processId": "1",     "vendor": "Oracle",     "application": "ocnssf",     "engVersion": "22.2.0.0.0",     "mktgVersion": "22.2.0.0.0",     "mktgVersion": "22.2.0.0.0",     "microservice": "nsavailability",     "namespace": "ocnssf",     "node_name": "k8s-     node-8.bulkhead.lab.us.oracle.com" }</pre>



#### **Maintain**

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Troubleshooting Guide*.
- 2. Raise a service request: See My Oracle Support for more information on how to raise a service request.

## 4.20 Handover from EPS to 5G

The current market retains a broad mix of UEs in both 4G and 5G networks. These UEs move from one network to another network in multiple ways:

- 1. From one 5G PLMN to another 5G PLMN (VPLMN), that is from one 5G network to another 5G network (also known as 5G to 5G roaming).
- 2. From 4G network to 5G network inside the same "PLMN" (also known as EPS to 5G).

In both the scenarios, a slice mapping mechanism is required on the NSSF of a visited PLMN (VPLMN). This feature implements the support for EPS to 5G slice mapping in NSSF, that is the second case in the list given above.

When a UE moves from 4G network to a 5G network (EPS to 5G), a registration of UE is triggered in the 5G network. The AMF, which caters to UE in 4G, requests to MAP and figure out Authorized 5G S-NSSAIs for the UE. However, the EPS to 5G move is only possible when there are converged 4G-5G nodes on operator network.

Following are high-level scenarios for EPS to 5G movement of a UE:

**Scenario 1:** UE is moving from EPS to 5G of a VPLMN (For example: Customer of Network-1 is moving from EPS of Network-2 to 5G of Network-2).

**Scenario 2:** UE is moving from EPS to 5G of the UE's HPLMN (For example: Customer of Network-1 is moving from EPS of Network-1 to 5G of Network-1).

#### (i) Note

- when UE is in roaming, the Get request in URI parameters contains home-plmn-id from table Table 6.1.3.2.3.1-1: 3GPP TS 29.531 version 16.3.0 Release 16.
- In the scenarios where home PLMN is not present, UE is not roaming.

The following call flow takes places in the entire process of EPS to 5G handover:

- 1. AMF sends a SliceInfoForRegistration GET request to NSSF. Only the following three parameters are considered for this feature when requestMapping is set to true:
  - sNssaiForMapping
  - requestedNssai
  - requestMapping
- 2. The query parameters may also contain:
  - Mapping to the Configured NSSAI for the HPLMN



- PLMN ID of the Subscription Permanent Identifier (SUPI)
- UE's current Tracking Area
- NF type of the NF service consumer
- AMF ID
- 3. NSSF identifies from the messages if AMF or UE requires EPS to 5G handover.
- 4. If yes, NSSF sends selected NSSAI based on below conditions:
  - **a.** If mapping is already provided, NSSF uses the mapped S-NSSAI and applies the policy.
  - b. If requestMapping is enabled, NSSF takes the 4G slice and maps it to 5G S-NSSAI by sending the corresponding NSSAI in the allowedNssaiList after policy check and confirms the same by responding with AuthorizedNetworkSliceInfo.
  - c. If the mapping is not available, the NSSF responds with 4XX status.

The following tables provide the details of the request (SliceInfoForRegistration) and the response (AuthorizedNetworkSliceInfo), respectively:

Table 4-7 Request: SliceInfoForRegistration

Attribute name	Data type	Description
sNssaiForMapping	array(Snssai)	This IE is included if the requestMapping IE is set to true. When included, the IE may contain any of the following:  The set of S-NSSAIs obtained from PGW+SMF in the HPLMN for PDU sessions that are handed over from EPS to 5GS.
		<ul> <li>The set of HPLMN S-NSSAIs obtained from source AMF during handover procedure within 5GS.</li> <li>The S-NSSAIs for the HPLMN received from the UE during EPS to 5GS idle mode Mobility Registration Procedure using N26 interface or idle state mobility registration procedure in 5GS.</li> </ul>
requestedNssai	array(Snssai)	<ul> <li>This IE contains the set of S-NSSAIs requested by the UE.</li> <li>During EPS to 5GS handover procedure using N26 interface, this IE contains the set of S-NSSAIs in the serving PLMN obtained from PGW+SMF in VPLMN or mapped from the set of S-NSSAIs obtained from PGW+SMF in the HPLMN.</li> <li>During handover procedure within 5GS, the IE contains the set of S-NSSAIs in the serving PLMN obtained from the source AMF, or mapped from the set of HPLMN S-NSSAIs obtained from source AMF.</li> </ul>
requestMapping	boolean	This IE may be present when the Nnssf_NSSelection_Get procedure is invoked during EPS to 5GS Mobility Registration Procedure (Idle State) using N26 interface or during EPS to 5GS handover procedure using N26 interface. This IE may also be present when Nnssf_NSSelection_Get procedure is invoked during idle state Mobility Registration Procedure or handover procedure in 5GS.
		When present this IE indicates to the NSSF that the NSSF returns the VPLMN specific mapped SNSSAI values for the S-NSSAI values in the sNssaiForMapping IE.



Table 4-8 Response: AuthorizedNetworkSliceInfo

Attribute name	Data type	Description
allowedNssaiList	array(AllowedNssai)	This IE is included one of the following conditions is true:
		The NSSF received the Requested NSSAI and the subscribed S-NSSAI(s).
		• The request Mapping flag in the corresponding request is set to true.
		When present, this IE may contain any of the following:
		The allowed S-NSSAI(s) authorized by the NSSF in the serving PLMN per access type if the Requested NSSAI and the Subscribed S-NSSAI(s) received.
		The mapping of S-NSSAI(s) of the VPLMN to corresponding HPLMN S-NSSAI(s) if requestMapping flag is set to true.
		NSSF considers load level information of a Network Slice Instance, provided by the NWDAF, to exclude slices that are overloaded.

#### Managing Handover from EPS to 5G

#### **Enable**

This feature is driven by 3GPP specifications. There is no option to enable or disable this feature. If all the **prerequisites** are met, it will be auto enabled at the time of installation or after upgrade to the target version.

#### Configure using REST API or CNC Console

Operator must configure MappingOfNssai for each PLMN for this feature to work.

For more information on REST based configuration, see **MappingOfNssai** in *Oracle* Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.

For more information on CNC Console based configuration, see Mapping of Nssai.

#### **Observe**

No new metrics or KPIs are generated for this feature. For information on other Metrics and KPIs of OCNSSF, see OCNSSF Metrics and OCNSSF KPIs sections respectively.

#### **Error Scenarios**

For the EPS to 5G Handover Feature, logs are generated for NSSelection mapping, when error is due to slice mapping determination or assignment to a UE or PDU session.



Table 4-9 Error Scenarios

Microservice	Details
NSSelection	Response Code/ Error Title:
	No mapping 5G SNSSAI found for SnssaiList in PLMN
	RequestMappingFailed
	403 Forbidden SNSSAI_NOT_SUPPORTED
	Log Snippet:
	<b>\{</b>
	"instant": {
	"epochSecond": 1661327119,
	"nanoOfSecond": 10456886
	<pre>}, "thread": "nf-mediation-thread-1",</pre>
	"level": "ERROR",
	"loggerName":
	"com.oracle.cgbu.cne.nssf.nsselection.service
	.NsPolicyServiceImpl",
	"message": "No mapping 5G snssai found
	for[3:EABB03, 4:EABB04]inPlmn [mcc=100,
	mnc=101]",
	"endOfBatch": false,
	"loggerFqcn":
	"org.apache.logging.log4j.spi.AbstractLogger"
	<pre>"contextMap": {},</pre>
	"threadId": 54,
	"threadPriority": 5,
	"ts": "2022-08-24 07:45:19.010+0000",
	"ocLogId": "\${ctx:ocLogId}",
	"pod": "ocnssf-nsselection-5bb7bb7799-
	gcs5r",
	"processId": "1",
	"vendor": "Oracle",
	"application": "ocnssf",
	"engVersion": "22.3.0", "mktgVersion": "22.3.0.0.0",
	"microservice": "nsselection",
	"namespace": "ocnssf",
	"node_name": "jazz-k8s-node-8"
	}



Table 4-9 (Cont.) Error Scenarios

Scenario	Microservice	Details
Mapping not found for one or more	NSSelection	Response Code/ Error Title:
S-NSSAIs and found for others.		NSSF logs error for S-NSSAI for which mapping is not found
		Log Snippet:
		<pre>"instant": {     "epochSecond": 1661327653,         "nanoOfSecond": 516132707 },     "thread": "nf-mediation-thread-2",     "level": "ERROR",     "loggerName": "com.oracle.cgbu.cne.nssf.nsselection.service .NsPolicyServiceImpl",     "message": "No mapping 5G snssai found for3:EABB03",     "endOfBatch": false,     "loggerFqcn": "org.apache.logging.log4j.spi.AbstractLogger" ,     "contextMap": {},     "threadId": 55,     "threadPriority": 5,     "ts": "2022-08-24 07:54:13.516+0000",     "ocLogId": "\${ctx:ocLogId}",     "pod": "ocnssf-nsselection-5bb7bb7799- gcs5r",     "processId": "1",     "vendor": "Oracle",     "application": "22.3.0",     "mktgVersion": "22.3.0.0.0",     "microservice": "nsselection",     "namespace": "ocnssf",     "node_name": "jazz-k8s-node-8" }</pre>



Table 4-9 (Cont.) Error Scenarios

Scenario	Microservice	Details
No allowed S-NSSAI are found for	NSSelection	Response Code/ Error Title:
accessType = 3GPP		RequestMappingFailed
		403 Forbidden SNSSAI_NOT_SUPPORTED
		Log Snippet:
		<pre>log simple:  {     "instant": {         "epochSecond": 1661327653,         "nanoOfSecond": 609737045 },      "thread": "nf-mediation-thread-2",     "level": "ERROR",     "loggerName":  "com.oracle.cgbu.cne.nssf.nsselection.service .NsPolicyServiceImpl",     "message": "No allowed snssai with access type 3GPP for request mapping=true. Throwing ForbiddenException.",     "endOfBatch": false,     "loggerFqcn":  "org.apache.logging.log4j.spi.AbstractLogger",  "contextMap": {},     "threadId": 55,     "threadPriority": 5,     "ts": "2022-08-24 07:54:13.609+0000",         "ocLogId": "\${ctx:ocLogId}",         "pod": "ocnssf-nsselection-5bb7bb7799- gcs5r",     "processId": "1",     "vendor": "Oracle",     "application": "ocnssf",     "engVersion": "22.3.0.0.0",     "mitrgVersion": "22.3.0.0.0",     "mitrgVersion": "22.3.0.0.0",     "microservice": "nsselection",     "namespace": "ocnssf",     "node_name": "jazz-k8s-node-8" }</pre>

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- **1. Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Network Slice Selection Function Troubleshooting Guide.*
- 2. Raise a service request: See My Oracle Support for more information on how to raise a service request.



## 4.21 Feature Negotiation

This feature negotiates optional features applicable between NSSF and NF Service Consumer (AMF/V-NSSF) for the NSSF supported services. The NF Service Consumer indicates the optional features it supports for the Nnssf NSSAIAvailability or Nnssf NSSElection service by including the supported feature attributes.

The following optional supported features are defined for NSSF as per 3GPP:

#### Nnssf NSSAlAvailability service supportedFeatures attributes:

- Subscription Modification (SUBMOD): This feature allows the operator to modify subscriptions by supporting HTTP Patch on NSAvailability (/nssai-availability/ subscriptions/). When Subscription Modification in Subscribe Service Operation (SUMOD) is supported, the operator can modify the subscription of NSSAI availability by implementing the HTTP Patch method.
- Empty Authorized NSSAI Availability Notification (EANAN): When this feature is supported, an NF Consumer that supports EANAN accepts an empty array of Authorized NSSAI Availability Data in a notification from NSSF and deletes locally stored Authorized NSSAI Availability Data that was received previously.
- Optimized NSSAI Availability Data Encoding (ONSSAI): ONSSAI is one of the optional features supported by NSSF. NSSF, this feature is described in 3GPP TS 29.531. When ONSSAI is supported by AMF and NSSF, NSSAI Availability data may be signaled per list or per range(s) of Tracking Area Identifiers(TAIs).

Supported Feature Information Element (IE): Supported Feature is a hexadecimal string that contains a bitmask indicating supported features. Each character in the string can take a value of "0" to "9", "a" to "f" or "A" to "F". The character representing the highest-numbered features appears first in the string, and the character representing features 1 to 4 appears last in the string. The list of features and their numbering (starting with 1) are defined separately for each API. If the string contains a lower number of characters, then there are defined features for an API.



#### Note

Features represented by the characters that are not present in the string are not supported.

**Table 4-10** SupportedFeatures for NSAvailability

Supported Feature based on supported feature set	ES3XX	EANAN	SUMOD	ONSSAI
"0"	no	no	no	no
"1"	no	no	no	yes
"2"	no	no	yes	no
"3"	no	no	yes	yes
"4"	no	yes	no	no
"5"	no	yes	no	yes
"6"	no	yes	yes	no
"7"	no	yes	yes	yes



Table 4-10 (Cont.) SupportedFeatures for NSAvailability

Supported Feature based on supported feature set	ES3XX	EANAN	SUMOD	ONSSAI
"8"	yes	no	no	no
"9"	yes	no	no	yes
"A"	yes	no	yes	no
"B"	yes	no	yes	yes
"C"	yes	yes	no	no
"D"	yes	yes	no	yes
"E"	yes	yes	yes	no
"F"	yes	yes	yes	yes

Table 4-11 SupportedFeatures for NSSelection

Supported Feature based on supported feature set	ES3XX
"0"	no
"1"	yes

#### **Managing Feature Negotiation**

#### **Enable**

To enable this feature, set the global.SupportedFeatureNegotiationEnable parameter to true under the global section in the ocnssf\_custom\_values\_25.1.100.yaml file.

The following snippet represents the location of the mentioned parameter in the Helm file:

#### global:

SupportedFeatureNegotiationEnable: true

#### Configure

There are no additional configurations required.

#### **Observe**

No new metrics or KPIs are generated for this feature. For information on other Metrics and KPIs of OCNSSF, see OCNSSF Metrics and OCNSSF KPIs sections respectively.



#### **Error Scenarios**

**Table 4-12 Error Scenarios** 

Scenario	Helm Configuration	Output
NSSelection Get with supported feature. That is, '1'	SupportedFeatureNegotiationEnable:" true" 3gppFeatures: NsSelection: ES3XX: "true" NsAvailability: ONSSAI: "true" SUMOD: "true" EANAN: "true" ES3XX: "true"	Response with supported feature i.e. '1'
NSSelection Get with supported feature. That is, '2'	SupportedFeatureNegotiationEnable:" true" 3gppFeatures: NsSelection: ES3XX: "true" NsAvailability: ONSSAI: "true" SUMOD: "true" EANAN: "true" ES3XX: "true"	Response without supported feature Unsupported value provided for the Supported Feature. Maximum supportedFeatured value is: 1
NSAvailability request with supported feature. That is, '2'	SupportedFeatureNegotiationEnable:" true" 3gppFeatures: NsSelection: ES3XX: "true" NsAvailability: ONSSAI: "true" SUMOD: "true" EANAN: "true" ES3XX: "true"	Response with supported feature i.e. '2'



Table 4-12 (Cont.) Error Scenarios

Scenario	Helm Configuration	Output
NSAvailability request with supported feature. That is, '2'	SupportedFeatureNegotiationEnable: " true" 3gppFeatures: NsSelection: ES3XX: "true" NsAvailability: ONSSAI: "true" SUMOD: "false" EANAN: "true" ES3XX: "true"	Bad request 400 Error: All requested supported features are not enabled on NSSF. Enable features from NSSF are:ONSSAI,EANAN,ES3XX
NSAvailability request with supported feature. That is, '7'	SupportedFeatureNegotiationEnable:" true" 3gppFeatures: NsSelection: ES3XX: "true" NsAvailability: ONSSAI: "false" SUMOD: "true" EANAN: "false" ES3XX: "false"	Bad Request 400 Error: All requested supported features are not enabled on NSSF. Enable features from NSSF are:SUMOD

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Network Slice Selection Function Troubleshooting Guide*.
- 2. Raise a service request: See My Oracle Support for more information on how to raise a service request.

## 4.22 Subscription Modification Feature

This feature allows operator to modify subscription by supporting HTTP Patch on NsAvailability subscribe service operation (/nssai-availability/subscriptions/). Supported operations on HTTP Patch are ADD, REMOVE, and REPLACE. Whereas, COPY, MOVE, and TEST are not supported.

#### **Managing Subscription Modification**

#### **Enable**

To enable this feature, set the global.SupportedFeatureNegotiationEnable and global.threegppFeatures.NsAvailability.SUMOD parameters to true under the global section in the ocnssf\_custom\_values\_25.1.100.yaml file.



The following snippet represents the location of the mentioned parameters in the Helm file:

```
global:
   SupportedFeatureNegotiationEnable: true
   threegppFeatures:
    NsAvailability:
   SUMOD: true
```

# Configure

There are no additional configurations required.

#### Observe

- Added the following success measurements:
  - ocnssf\_nssaiavailability\_submod\_rx\_total
  - ocnssf\_nssaiavailability\_submod\_success\_response\_tx\_total
- Added the following error measurements:
  - ocnssf\_nssaiavailability\_submod\_error\_response\_tx\_total
  - ocnssf\_nssaiavailability\_submod\_unimplemented\_op\_total
  - ocnssf\_nssaiavailability\_submod\_patch\_apply\_error\_total

For information about other Metrics and KPIs of NSSF, see <u>NSSF Metrics</u> and <u>NSSF KPIs</u> sections respectively.

#### **Error Scenarios**

Table 4-13 Error Scenarios

Scenario	Microservice	Description
Patch request processing failed due	Nnssf_NSSAIAvailabilit	Request URL:
to invalid path	у	nnssf-nssaiavailability/v1/nssai-availability/subscriptions/
		Response Code/ Error Title:
		400
		400 Bad Request
		Error Jason Patch Req processing failed
Subscription with HTTP Patch	Nnssf_NSSAlAvailabilit	Request URL:
(Option ADD). Subscription present and TAI addition not supported in PLMN.	У	nnssf-nssaiavailability/v1/nssai-availability/subscriptions/
		Response Code/ Error Title:
		HTTP 403
		Unsupported PLMN
		Error Details must specify supported PLMN list
SUBMOD is set to false	Nnssf_NSSAlAvailabilit	Request URL:
	у	nnssf-nssaiavailability/v1/nssai-availability/subscriptions/
		Response Code/ Error Title:
		HTTP 405
		Method not allowed



# Table 4-13 (Cont.) Error Scenarios

Scenario	Microservice	Description
Subscription ID is not found	Nnssf_NSSAIAvailabilit	Request URL:
	У	nnssf-nssaiavailability/v1/nssai-availability/subscriptions/
		Response Code/ Error Title:
		HTTP 404
		Not Found

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Troubleshooting Guide*.
- **2.** Raise a service request: See My Oracle Support for more information on how to raise a service request.

# 4.23 Empty Authorized NSSAI Availability Notification

Empty Authorized NSSAI Availability Notification (EANAN) feature provides support for sending an empty array of Authorized NSSAI Availability Data when a notification trigger leads to a situation of no Tracking Area (TA) with Authorized NSSAI by the NSSF. An Access and Mobility Management Function (AMF) that supports this feature accepts the empty array of Authorized NSSAI Availability Data in a notification from NSSF and deletes locally stored Authorized NSSAI Availability Data that was received previously.

## **Managing EANAN**

#### **Enable**

To enable this feature, set the global.SupportedFeatureNegotiationEnable and global.threegppFeatures.NsAvailability.EANAN parameters to true under the global section in the ocnssf\_custom\_values\_25.1.100.yaml file.

The following snippet represents the location of the mentioned parameters in the Helm file:

```
global:
   SupportedFeatureNegotiationEnable: true
   threegppFeatures:
   NsAvailability:
      EANAN: true
```

# Configure

There are no additional configurations required.

## Observe

No new metrics or KPIs are generated for this feature. For information on other Metrics and KPIs of OCNSSF, see OCNSSF Metrics and OCNSSF KPIs sections respectively.



# **Scenarios**

Table 4-14 Scenarios

Scenario	Helm Configuration	Details	
Send empty notification when EANAN is supported by both NSSF and Consumer NF for delete as notification trigger	<pre>global.SupportedFeatureNegotiationEnable : true global.threegppFeatures.NsAvailability.E ANAN : true</pre>	Subscription:	
		Configure nssai_auth to allow     S-NSSAI-1 in TAI-1	
		Send a subscribe for TAI-1 with supportedFeatures flag with EANAN bit set to true	
		3. Delete nssai_auth	
		Output:	
		Configuration of nssai_auth     must be successful	
		Subscription response must contain Authorized NSSAI Availability Data as S-NSSAI-1 for Tai-1 with EANAN bit in supportedFeaturesset to true	
		a. Delete nssai_auth must be done	
		b. Notification from NSSF must contain empty Authorized NSSAI Availability Data	
Do not send empty notification	global.SupportedFeatureNegotiationEnable	Subscription:	
when EANAN is supported by NSSF and not by Consumer NF	: true  global.threegppFeatures.NsAvailability.E  ANAN : true	Configure nssai_auth to allow     S-NSSAI-1 in TAI-1	
		Send a subscribe for TAI-1 with supportedFeatures flag with EANAN bit set to false	
		3. Delete nssai_auth	
		Output:	
		Configuration of nssai_auth     must be successful	
		Subscription response must contain Authorized NSSAI Availability Data as S-NSSAI-1 for Tai-1 with EANAN bit in supportedFeaturesset to false	
		3. Delete nssai_auth must be done.	



Table 4-14 (Cont.) Scenarios

Scenario	Helm Configuration	De	tails
Do not send empty notification when EANAN is not supported by NSSF and supported by Consumer NF	global.SupportedFeatureNegotiationEnable : true	Su	bscription:
		1.	Configure nssai_auth to allow S-NSSAI-1 in TAI-1
	global.threegppFeatures.NsAvailability.E ANAN : false	2.	Send a subscribe for TAI-1 with supportedFeatures flag with EANAN bit set to true
		3.	Delete nssai_auth
		Ou	tput:
		1.	Configuration of nssai_auth must be successful
		2.	Subscription response must contain Authorized NSSAI Availability Data as S-NSSAI-1 for Tai-1 with EANAN bit in supportedFeaturesset to false
		3.	Delete nssai_auth must be done
Do not send empty notification	global.SupportedFeatureNegotiationEnable	Su	bscription:
when EANAN is not supported by NSSF and supported by Consumer NF	: true  global.threegppFeatures.NsAvailability.E  ANAN : false	1.	Configure nssai_auth to allow S-NSSAI-1 in TAI-1
		2.	Send a subscribe for TAI-1 with supportedFeatures flag with EANAN bit set to true
		3.	Delete nssai_auth
		Ou	tput:
		1.	Configuration of nssai_auth must be successful
		2.	Subscription response must contain Authorized NSSAI Availability Data as S-NSSAI-1 for Tai-1 with EANAN bit in supportedFeaturesset to false Delete nssai_auth must be
		J.	done

# Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Network Slice Selection Function Troubleshooting Guide*.
- 2. Raise a service request: See My Oracle Support for more information on how to raise a service request.



# 4.24 Optimized NSSAI Availability Data Encoding and TAI Range

Support for TAI range and Optimized NSSAI Availability Data Encoding(ONSSAI) is introduced in NSSF as per 3GPP TS 29.531 (Release 16):

NSSF supports Optimized NSSAI Availability Data Encoding (ONSSAI), which further extends support for TAI ranges. These two features work together to provide multiple benefits to NSSF. Listed below are the benefits of this feature:

## Support for ONSSAI as per 3GPP Specifications 29.531 release 16

ONSSAI is one of the 3GPP Specification 29.531 supported optional features that NSSF negotiates using <u>Feature Negotiation</u>. When ONSSAI is supported by AMF and NSSF, NSSAI Availability data may be signaled per list or per range(s) of Tracking Area Identifiers(TAIs).

The feature bit of ONSSAI is exchanged between NSSF and NF Service Consumer, indicating the support for taiList and taiRangeList. With this support, NSSF and NF Service Consumers can exchange the capabilities in which the support of taiList and taiRangeList is also communicated. NSSF can now expose APIs and store the supported NSSAIs with respect to taiList and taiRangeList. This enables the operator to provide data with minimal entries, which reduces the effort while system configuration.

# Support for TAI range in NSSF.

With the advent of 5G and the three types of 5G network slices (MIoT, eMBB, and URLLC), there could be millions of TAIs ( and TACs) in a PLMN. As per current standards, the maximum number of slice instances in a 5G network can reach hundreds. Although the number of slice identifiers (SNSSAIs) in a PLMN is operator-dependent, there is a scope to scale this in thousands, and corresponding slice instance (NSI) in tens of thousands. Since there is a many-to-many mapping between TAIs and SupportedSNSSAIs, and this information is shared and authorized between NSSF and NF Service Consumer of nssai\_availability service, the number of TAIs supported by an AMF can range in millions. This can lead to a massive increase in the number of messages with replicated data across NSSF and AMF.

The support for TAI ranges reduces the memory consumption at NSSF by eliminating the need for redundant data storage. Apart from this, the size of supported NSSAI towards NF Service Consumer is also reduced, which further improves the overall network throughput.

## For Example:

Consider a case with PLMN 100,101, which constitutes of 1000000 TACs; (1,10,00,000).

If the AMF Supports 300000 TACs, for TAC (1 - 300000):

- Supported SNSSAIs in TAC (1 -100000) are SNSSAI-1,SNSSAI-2,SNSSAI-3,
- supported SNSSAIs in TAC (100001 200000) are SNSSAI-2, SNSSAI-3, SNSSAI-4, and
- supported SNSSAIs in TAC (200001 300000) are SNSSAI-3, SNSSAI-4, SNSSAI-5.

Each AMF supports multiple TAIs and S-NSSAIs, as shown in the diagram below:



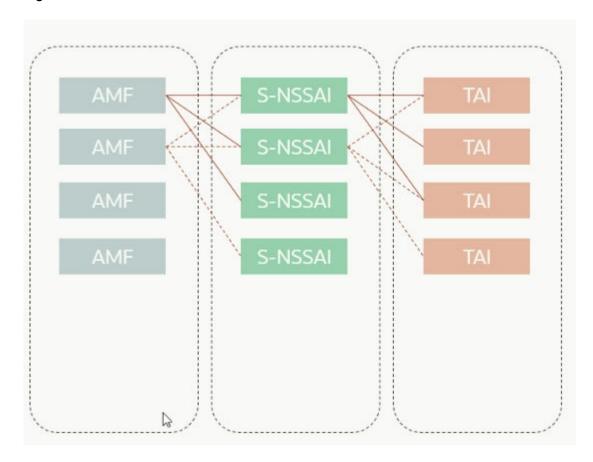


Figure 4-2 Association of AMF and TAI with S-NSSAI

The list of supported TAIs will be almost the same, but for different S-NSSAIs, NSSF requires huge storage space for the redundant data. This results in a substantial increase in the size of the notifications and hassles for managing such a huge configuration, as explained below:

# Case 1: nssaiAvailabilityInfo without ONSSAI

Here, the list size of supportedNssaiAvailabilityData would be 300,000 with 3 unique and 299997 repeated SNSSAI list, here is a sample snippet:



```
"sst": 1
                     "sd": "000003",
                     "sst": 1
            ]
        },
{
             "tai": {
                 "plmnId": {
                     "mcc": "100",
                     "mnc": "101"
                 },
                 "tac": "000002"
            },
             "supportedSnssaiList": [
                     "sd": "000001",
                     "sst": 1
                 },
                     "sd": "000002",
                     "sst": 1
                 },
                     "sd": "000003",
                     "sst": 1
            1
        }...the list will go on for 3 unique and 299997 repeated SNSSAI list.
    ]
}
```

# Case 2: nssaiAvailabilityInfo with ONSSAI

With the support for TAI range, there is an overlapping of TAIs. So, a TAI list of size 3 is enough to contain all 300,000 SNSSAI lists.

Following is a sample snippet:



```
"sd": "000002",
             "sst": 1
        },
             "sd": "000003",
             "sst": 1
    ],
    "taiRangeList":[
             "plmnId": {
             "mcc": "100",
             "mnc": "101"
        "tacRangeList":[
             "start": "000001",
             "end": "100000"
             ]
        ]
},
    "tai": {
        "plmnId": {
             "mcc": "100",
             "mnc": "101"
        "tac": "000001"
    },
    "supportedSnssaiList": [
             "sd": "000002",
             "sst": 1
            "sd": "000003",
             "sst": 1
             "sd": "000004",
             "sst": 1
    ],
    "taiRangeList":[
             "plmnId": {
             "mcc": "100",
             "mnc": "101"
        "tacRangeList":[
             "start": "100001",
```



```
"end": "200000"
                     ]
                 ]
             "tai": {
                 "plmnId": {
                     "mcc": "100",
                     "mnc": "101"
                 "tac": "000001"
            },
             "supportedSnssaiList": [
                     "sd": "000003",
                     "sst": 1
                     "sd": "000004",
                     "sst": 1
                     "sd": "000005",
                     "sst": 1
            ],
             "taiRangeList":[
                     "plmnId": {
                     "mcc": "100",
                     "mnc": "101"
                 },
                 "tacRangeList":[
                     "start": "200001",
                     "end": "300000"
                     ]
        }
    ]
}
```

# Managing ONSSAI and TAI Range

# Enable

To enable ONSSAI, set the global.SupportedFeatureNegotiationEnable and global.threegppFeatures.NsAvailability.ONSSAI parameter to true under the global section in the ocnssf\_custom\_values\_25.1.100.yaml file.



The following snippet from the yaml file represents the location of the mentioned parameters in the Helm file:

```
global:
   SupportedFeatureNegotiationEnable: true
   threegppFeatures:
   NsAvailability:
    ONSSAI: true
```

#### Configure

# **Configuration using Helm Parameters:**

There are no additional configurations required in the helm parameters.

# **Configuration using REST API:**

TacRange managed object, and tacrange parameter under NSSAI Auth and NSS Rules provide the support required to use TAI range feature. To perform the feature configuration, see NssRule, NssaiAuth, and TacRange managed objects in the chapter NSSF Managed Objects of Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.

#### **Configuration using CNC Console:**

For more information, see NSS Rule and NSSAI Auth.

#### Observe

No new metrics or KPIs are generated for this feature. For information on other Metrics and KPIs of OCNSSF, see OCNSSF Metrics and OCNSSF KPIs sections respectively.

# Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Network Slice Selection Function Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

# 4.25 Georedundancy

NSSF supports up to three-site Georedundancy to ensure service continuity when one of the NSSF sites is down. When NSSF is deployed as georedundant NSSF instances, then:

- All the sites that register with NRF work independently and are in Active state.
- Based on the Rank, each NSSF site subscribes to NRF for any state change of other NSSF sites and gets notified when an NSSF site goes down.
- The NFs in a given site need to configure one of the georedundant NSSF as the primary NSSF and others as secondary NSSF and tertiary NSSF, respectively.
- When the primary NSSF is available, the NFs send service requests to the primary NSSF.
   When the NSSF at the primary site is unavailable, the NFs redirect service requests to the secondary NSSF or tertiary NSSF, until the primary NSSF's Active status is restored.



Priority based NSSF selection (at NF Consumer or SCP) can be implemented to ensure route traffic based on which NSSF site is up.

The NSSF's data gets replicated between the georedundant sites by using DB tier's replication service.

With NSSF georedundant feature, the NSSF Services (NSSelection and NSAvailability) will continue to work as independent service operations.

Following are the prerequisites for georedundancy:

- Each site must configure remote NSSF sites as georedundant mates.
- The configurations at each site must be same. The NSSF at all sites must handle the NFs in the same manner.
- Once the Georedundancy feature is enabled on a site, it cannot be disabled.
- If the Time Of the Day (TOD) feature is enabled, georedundant sites are time synchronized.
- NFs need to configure georedundant NSSF details as Primary, Secondary, and Tertiary NSSFs.
- Georedundant sites must have REST based configuration as explained in Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.
- At any given time, NFs must communicate with only one NSSF. That is, NFs must register services and maintain heartbeats with only one NSSF. The data must be replicated across the georedundant NSSFs, allowing seamless NF mobility across NSSFs as required.

#### **Managing NSSF Georedundancy Feature**

# **Prerequisites**

Following are the prerequisites to enable georedundancy feature in NSSF:

- cnDBTier must be installed and configured for each site. For the installation procedure, see Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide.
- The Database Replication Channels between the sites must be up.
- Configure MySQL Database, Users and Secrets. For the configuration procedure, see Preinstallation Tasks in Oracle Communications Cloud Native Core, Network Slice Selection Function Installation an Upgrade Guide.

# **Enable Georedundancy Feature**

Configure the following to enable the georedundancy feature:



# (i) Note

Configuring these attributes during deployment is mandatory before enabling the georedundancy feature. Otherwise, georedundancy cannot be enabled, and NSSF at the site will act as a stand-alone NSSF.

# **Helm Configuration for Database:**



Configure the following parameters in the ocnssf\_custom\_values\_25.1.100.yaml file for all three sites:

- global.stateDbName
- global.provisionDbName
- global.releaseDbName
- global.nameSpace
- global.mysql.primary.host

#### At Site 1:

```
global:
# Mysql NSSF Database Names
 stateDbName: 'nssfStateDB'
 provisionDbName: &provDB 'nssfProvSite1DB'
  # Mysql Release Database Name used to maintain release version
  releaseDbName: 'ocnssfReleaseDB'
  # NameSpace where secret is deployed
  nameSpace: &ns ocnssf1
  # Database configuration
  mysql:
   primary:
     host: &dbHost "mysql-connectivity-service.sitel"
At Site 2:
```

#### qlobal:

```
# Mysql NSSF Database Names
 stateDbName: 'nssfStateDB'
provisionDbName: &provDB 'nssfProvSite2DB'
 # Mysql Release Database Name used to maintain release version
 releaseDbName: 'ocnssfRelease2DB'
 # NameSpace where secret is deployed
 nameSpace: &ns ocnssf2
 # Database configuration
 mysql:
   primary:
     host: &dbHost "mysql-connectivity-service.site2"
```



#### At Site 3:

```
global:

# Mysql NSSF Database Names
   stateDbName: 'nssfStateDB'
   provisionDbName: &provDB 'nssfProvSite3DB'

# Mysql Release Database Name used to maintain release version releaseDbName: 'ocnssfRelease3DB'

# NameSpace where secret is deployed nameSpace: &ns ocnssf3

# Database configuration mysql:
   primary:
   host: &dbHost "mysql-connectivity-service.site3"
```

## **Helm Configuration of Parameters:**

Configure the following parameters in the ocnssf\_custom\_values\_25.1.100.yaml file for all three sites:

- global.grEnabled
- global.nfInstanceId
- global.siteId
- If global.grEnabled is set to true, Configure the following parameters as well:
  - global.grEnv.maxSecondsBehindRemote
  - global.grEnv.dbMonitorServiceUrl
  - global.grEnv.peerGRSitesList.siteId
  - global.grEnv.peerGRSitesList.nfInstanceId

For more information about configuring the parameters, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Installation*, *Upgrade*, *and Fault Recovery Guide*.

Following is the sample configuration at Site named "site1" (nfInstanceId: 9faf1bbc-6e4a-4454-a507-aef01a101a01), which is georedundant with Sites, "site2" (NssfInstanceId: 9faf1bbc-6e4a-4454-a507-aef01a101a02) and "site3" (NssfInstanceId: 9faf1bbc-6e4a-4454-a507-aef01a101a03):



```
#All parameters under this section are valid only if grEnabled is true
  arEnv:
    #Maximum allowed seconds behind remote site for replication
    maxSecondsBehindRemote: 5
    #URL to check db-replication status
    dbMonitorServiceUrl: "http://mysql-cluster-db-monitor-svc.sitel:8080/db-
tier/status/replication/realtime"
    #GR sites list
   peerGRSitesList:
      - siteId: "site2"
      - nfInstanceId: "9faf1bbc-6e4a-4454-a507-aef01a101a02"
      - siteId: "site3"
      - nfInstanceId: "9faf1bbc-6e4a-4454-a507-aef01a101a03"
Following is the sample configuration at Site named "site2" (NssfInstanceId:
9faf1bbc-6e4a-4454-a507-aef01a101a02), which is georedundant with Sites, "site1"
(NssfInstanceId: 9faf1bbc-6e4a-4454-a507-aef01a101a01) and "site3" (NssfInstanceId:
9faf1bbc-6e4a-4454-a507-aef01a101a03):
global:
   #Only applicable for NSSF microservices
  # GR params
  #tag to enable GR
  grEnabled: true
  #InstanceId of NSSF used in case of GR
 nfInstanceId: "9faf1bbc-6e4a-4454-a507-aef01a101a02"
  #SiteID of NSSF used in case of GR
  siteId: "site2"
  #All parameters under this section are valid only if grEnabled is true
  qrEnv:
    #Maximum allowed seconds behind remote site for replication
    maxSecondsBehindRemote: 5
    #URL to check db-replication status
    dbMonitorServiceUrl: "http://mysql-cluster-db-monitor-svc.site2:8080/db-
tier/status/replication/realtime"
    #GR sites list
   peerGRSitesList:
      - siteId: "site1"
      - nfInstanceId: "9faf1bbc-6e4a-4454-a507-aef01a101a01"
      - siteId: "site3"
      - nfInstanceId: "9faf1bbc-6e4a-4454-a507-aef01a101a03"
Following is the sample configuration at Site named "site3" (NssfInstanceId:
9faf1bbc-6e4a-4454-a507-aef01a101a03), which is georedundant with Sites, "site1"
(NssfInstanceId: 9faf1bbc-6e4a-4454-a507-aef01a101a01) and "site2" (NssfInstanceId:
9faf1bbc-6e4a-4454-a507-aef01a101a02)
global:
  # GR params
  #tag to enable GR
  grEnabled: true
  #InstanceId of NSSF used in case of GR
  nfInstanceId: "9faf1bbc-6e4a-4454-a507-aef01a101a03"
  #SiteID of NSSF used in case of GR
```



```
siteId: "site3"
grEnv:
    #Maximum allowed seconds behind remote site for replication
    maxSecondsBehindRemote: 5
    #URL to check db-replication status
    dbMonitorServiceUrl: "http://mysql-cluster-db-monitor-svc.site3:8080/db-
tier/status/replication/realtime"
    #GR sites list
    peerGRSitesList:
        - siteId: "site1"
        - nfInstanceId: "9faf1bbc-6e4a-4454-a507-aef01a101a01"
        - siteId: "site2"
        - nfInstanceId: "9faf1bbc-6e4a-4454-a507-aef01a101a02"
```

# Observe

No new metrics or KPIs are generated for this feature. For information on other Metrics and KPIs of OCNSSF, see OCNSSF Metrics and OCNSSF KPIs sections respectively.

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

# 4.26 Time of the Day Based Network Slice Instance Selection

5G Traffic rate is not constant at all times, there might be spike in the traffic rate at certain busy hours. There can be occasional spikes based on holiday season. To manage these spikes and avoid traffic loss or over utilization of resources, operators may create network slice instance on need basis, or operators may create specific additional network slices to manage spikes in busy hours. The NSSF feature TOD (time of day based slice selection) allows operator to configure policy to select slice based on the time spans. Operator can provide date spans, day spans, time spans, or any combination of above to provide a policy to select network slice different catering to same SNSSAI based on date. This makes operator able to provide additional slices and enhances NSSF to select different slices based on another parameters. (apart from TAI + SNSSAI).

#### Managing Time of the Day Feature

#### **Enable**

This feature is enabled by default. If time spans are configured, NSSF selects network slice based on local NSSF. time.

#### Configure

#### Using helm parameters

If reqnftime flag is set to true, the time provided in Requester-NF-Time will the time considered. If reqnftime flag is set to false, local NSSF time will be considered.

# **Using Managed Objects**



This sample configuration shows a way to configure 2 time profiles and a rule to ensure time based selection of network slice instance profile.

Table 4-15 Sample configuration

Time profile	Network Slice instance profile
Week day rush hour	NSI-PROFILE-1
Christmas	NSI-PROFILE-2
Default fall back	NSI-PROFILE-3

# **Sample Time Profiles**

The following sample allows operator to configure a time profile for Weekdays busy hours. For more information on time profile configuration, see *Oracle Communication Cloud Native Core, Network Slice Selection Installation, Upgrade, and Fault Recovery Guide*.

```
"name": "WEEKDAY_BUSY",
"startDate": "2019-01-01",
"endDate": "2020-12-31",
"daysOfWeek": [
    "MONDAY",
    "TUESDAY",
    "WEDNESDAY",
    "THURSDAY",
    "FRIDAY"
],
"timeSpans": [
        "startTime": "07:00:00",
        "endTime": "12:00:00"
        "startTime": "17:00:00",
        "endTime": "22:00:00"
]
"name": "CHRISTMAS-DAY",
"startDate": "2019-12-24",
"endDate": "2019-12-25",
"daysOfWeek": [],
"timeSpans": []
```

The following time span enables user to configure a time profile for Christmas irrespective of day.

```
{
    "name": "CHRISTMAS-DAY",
    "start Date": "2019-12-24",
    "end Date": "2019-12-25",
```



```
"weekday": [],
"time Spans": []
```

Configure NSSF rule to select different profiles based on time of day

```
"name": "IR-RULE-TOD",
"amfId": "22345678-abcd-efAB-CDEF-123456789012",
"plmnId":
    "mcc": "102",
    "mnc": "102"
"tac": "100002",
"snssai":
    "sst": "1",
    "sd": "EABB01"
"salience": "0",
"behavior":
    "accessType": "3GPP_ACCESS",
    "nsiProfiles":
    ſ
        {
            "name": "NSI-PROFILE-2",
            "timeProfile": "CHRISTMAS DAY",
            "salience": 3
            "name": "NSI-PROFILE-1",
            "timeProfile": "WEEKDAY_BUSY",
            "salience": 2
            "name": "NSI-PROFILE-3",
            "salience": 1
    ]
},
```

#### (i) Note

In above rule configuration of *Salience* field for each time profile is to ensure that Christmas profile gets highest priority ( "salience": 3), then Week day rush hour and then fallback which is default profile.

Network slice profiles must be preconfigured. To configure slice profiles see, *Oracle Communication Cloud Native Core, Network Slice Selection Installation, Upgrade, and Fault Recovery Guide.* 



#### **Observe**

```
Following are the metrics related to Time of the Day feature : ocnssf_nsselection_rx_total ocnssf_nsselection_success_tx_total ocnssf_nsselection_policy_match_total ocnssf_nsselection_time_match_total
```

ocnssf nsselection nsi selected total

For further information about the Metrics and KPIs, see <u>NSSF Metrics</u> and <u>NSSF KPIs</u> sections respectively.

#### Maintain

- There must be a default fall back while configuring rule if none of the time profiles match.
- Salience of time profile within behavior section must be different for different time profiles; this it to have proper behavior if there is an overlap of time spans.

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- Collect the logs: For more information on how to collect logs, see Oracle Communications
   Cloud Native Core, Network Slice Selection Function Troubleshooting Guide.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

# 4.27 Multiple PLMN Support

This feature enables single NSSF instance to cater to multiple PLMNs. This enables operator to define slice selection policies for multiple PLMNs, and gives the option for operator to span a network slice across PLMNs.

NSSF allows the user to add the supported PLMN list which must be used for registering with NRF. Any change in supported PLMN list must trigger a Register request towards NRF with updated profile. Requests which have TAI containing other PLMN will be treated as roaming.

## **Managing Multiple PLMN Support**

#### **Enable**

If the global.supportedPlmnList has valid parameter values, then multiple PLMN feature is enabled.

To enable this feature, open the <code>ocnssf\_custom\_values\_25.1.100.yaml</code> file and set <code>global.supportedPlmnList</code> to valid values as shown the example below:

```
#Sample to enable
#Multiple PLMN support Following is the way to ensure only (100,101) and
(100,02)
  supportedPlmnList:
   - mcc: 100
     mnc: 101
   - mcc: 100
     mnc: 02
```



#### Disable

To disable this feature, open the ocnssf\_custom\_values\_25.1.100.yaml file and set global.supportedPlmnList to empty in Helm file, as shown the example below:

```
#Sample to disable multiple PLMN
supportedPlmnList: []
```



#### (i) Note

If the global.supportedPlmnList is empty, then the multiple PLMN feature is turned off, indicating that all PLMNs are permitted

#### **Observe**

Following are the metrics related to Multiple PLMN Support: ocnssf nsselection unsupported plmn total

ocnssf\_nsavailability\_unsupported\_plmn\_total

ocnssf\_nsselection\_rx\_total

ocnssf nsselection success tx total

ocnssf\_nsselection\_policy\_match\_total

ocnssf\_nsselection\_time\_match\_total

ocnssf\_nsselection\_nsi\_selected\_total

ocnssf\_nsselection\_policy\_not\_found\_total

For further information about the Metrics and KPIs, see NSSF Metrics and NSSF KPIs sections respectively.

**Error Scenarios** 

Table 4-16 Error Scenarios

Scenario	Microservice	Details
Request comes from unknown PLMN	NSSelection	Request URL: /nnssf-nsselection/v1/network-slice-information/ Response Code / Error Title:
		403 - PLMN_NOT_SUPPORTED  Notes: No query sent to DB. Look into configured PLMNs
		and respond
Request comes from known PLMN	NSSelection	Request URL: /nnssf-nsselection/v1/network-slice-information/ Response Code / Error Title: 200 OK based on policy match



Table 4-16 (Cont.) Error Scenarios

Scenario	Microservice	Details
Subscription request for unknown	NSAvailability	Request URL:
PLMNs only		/nnssf-nssaiavailability/v1/nssai-availability/subscriptions
		Response Code / Error Title:
		403 - PLMN_NOT_SUPPORTED
		Notes: No query sent to DB as none of the PLMNs are supported
Subscription request for unknown	NSAvailability	Request URL:
PLMNs and known PLMNs		/nnssf-nssaiavailability/v1/nssai-availability/subscriptions
		Response Code / Error Title:
		403 - PLMN_NOT_SUPPORTED
		<b>Notes:</b> No query sent to DB as none of the PLMNs are supported
AMF tries to store session data for	NSAvailability	Request URL:
unknown PLMNs only		nnssf-nssaiavailability/v1/nssai-availability
		Response Code / Error Title:
		403 - PLMN_NOT_SUPPORTED
		Notes: No query sent to DB as none of the PLMNs are supported
AMF tries to store session data for	NSAvailability	Request URL:
unknown PLMNs and known PLMNs		nnssf-nssaiavailability/v1/nssai-availability
LIVIIVO		Response Code / Error Title:
		403 - PLMN_NOT_SUPPORTED
		<b>Notes:</b> No query sent to DB as none of the PLMNs are supported
AMF tries to update session data	NSAvailability	Request URL:
for unknown PLMNs only		nnssf-nssaiavailability/v1/nssai-availability
		Response Code / Error Title:
		403 - PLMN_NOT_SUPPORTED
		<b>Notes:</b> No query sent to DB as none of the PLMNs are supported
AMF tries to update session data	NSAvailability	Request URL:
for unknown PLMNs and known PLMNs		nnssf-nssaiavailability/v1/nssai-availability
PLIVINS		Response Code / Error Title:
		403 - PLMN_NOT_SUPPORTED
		<b>Notes:</b> No query sent to DB as none of the PLMNs are supported
Operator tries to configure nsi_auth	NSConfig	Request URL:
for unsupported PLMN		/nnssf-configuration/v1/nssaiauth/ /nnssf-configuration/v1/nssrules/ /nnssf-configuration/v1/configurednssais
		Response Code / Error Title:
		403 - PLMN_NOT_SUPPORTED
		<b>Notes:</b> Currently, as we are not supporting the roaming scenario, the operator must not be allowed to add policy configuration for unknown PLMNs.

Here's the updated table with the merged columns under the new "Details" column:



#### **Maintain**

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Network Slice Selection Function Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

# 4.28 Support Indirect Communication

3GPP TS 29.531 Release 16 has introduced a new NF SCP which enables reliability and resiliency within network.

In indirect mode of communication consumers and producers interact through SCP. There are two communication models as described below:

**Model C - Indirect communication without delegated discovery:** Consumers do discovery by querying the NRF. Based on discovery result, the consumer does the selection of an NF Set or a specific NF instance of NF set. The consumer sends the request to the SCP containing the address of the selected service producer pointing to a NF service instance or a set of NF service instances. In the later case, the SCP selects an NF Service instance. If possible, the SCP interacts with NRF to get selection parameters such as Location, capacity, etc. The SCP routes the request to the selected NF service producer instance.

**Model D - Indirect communication with delegated discovery:** Consumers do not perform any discovery or selection. The consumer adds any necessary discovery and selection parameters required to find a suitable producer to the service request. The SCP uses the request address and the discovery and selection parameters in the request message to route the request to a suitable producer instance. The SCP can perform discovery with an NRF and obtain a discovery result

Once this feature is enabled on NSSF, it allows consumer NFs (AMF) to perform routing and rerouting to NSSF through SCP leveraging following 3GPP headers "3gpp-Sbi–Binding" and "3gpp-Sbi-Routing-Binding".





# Note

- This feature's scope involves the manipulation and updating of headers and values.
- It does not mandate that Notifications must go through SCP. The responsibility for configuring SCP to route the notifications is on the operator. For more information, see <u>DNS SRV Based Selection of SCP in NSSF</u>.
- NSSF only supports the following pattern of 3gpp-Sbi-Binding Header: bl=nf-set;

nfset=set<setId>.region<regionId>.amfset.5qc.mnc<mnc>.mcc<mcc>

 NSSF only accepts subscription with 3gpp-Sbi-Binding from AMF, provided AMF must be a part of the AMF-Set.

# **Managing Indirect Communication**

#### **Enable**

To enable this feature, set the value of indirectCommunicationSupportEnable to true in the ocnssf\_custom\_values\_25.1.100.yaml file.

```
# Indirect communication support
indirectCommunicationSupportEnable: true
```

The scope of this feature is Subscription and Notification flow.

When this feature is enabled:

 When AMF sends a NsAvailability Subscribe with 3gpp-Sbi-Binding header, NSSF validates if the header Supported Format is:

```
bl=nf-set;
nfset=set<setId>.region<regionId>.amfset.5gc.mnc<mnc>.mcc<mcc>
```

Only the following format is supported:

```
bl=nf-set;
nfset=set<setId>.region<regionId>.amfset.5gc.mnc<mnc>.mcc<mcc>
```

- In case of a successful validation, the NSSF responds with a 201 status code. The response includes a "3gpp-Sbi-Binding" header containing NSSF's binding information.
- The NSSF computes its binding information by matching the NSSF set details for the corresponding PLMN.
- The NSSF stores the Binding header of the AMF set in the database.
- When sending a notification for the subscription, the same value from the AMF's binding header is included in the notification as the "3gpp-Sbi-Routing-Binding" header.
- Additionally, the NSSF adds a "3gpp-Sbi-Callback" header with the value Nnssf\_NSSAIAvailability\_Notification.



- If the AMF sends a NsAvailability Subscribe request without the "3gpp-Sbi-Binding" header:
  - The NSSF responds without including the "3gpp-Sbi-Binding" header in the response.
  - The NSSF does not add the "3gpp-Sbi-Routing-Binding" header in the notification.
  - The processing of the request remains unchanged, and there is no impact on the processing and response, except that the mentioned headers are not computed or included as specified above.

#### Disable

To disable this feature, set the value of indirectCommunicationSupportEnable to false in the ocnssf\_custom\_values\_25.1.100.yaml file.

- When indirectCommunicationSupportEnable is set to false:
  - When AMF sends a NsAvailability Subscribe with 3gpp-Sbi-Binding header:
    - NSSF ignores the header and process the request.
    - \* NSSF does not add 3gpp-Sbi-Routing-Binding header in the notification.

# Indirect communication support
indirectCommunicationSupportEnable: false

#### **Observe**

- Added the following success measurements:
  - ocnssf nssaiavailability indirect communication rx total
  - ocnssf\_nssaiavailability\_indirect\_communication\_tx\_total
  - ocnssf nssaiavailability notification indirect communication tx total
  - ocnssf nssaiavailability notification indirect communication rx total
- Added the following error measurements:
  - ocnssf nssaiavailability indirect communication subscription failure total
  - ocnssf\_nssaiavailability\_indirect\_communication\_notification\_failure\_total

For more information on above metrics and KPIs, see <u>NSSF Metrics</u> and <u>NSSF KPIs</u>.



# **Error Scenarios**

**Table 4-17 Error Scenarios** 

Scenario	Input Details	Output
Subscription with binding,	Input message:	Subscription Response
global.indirectCommunicationSupp ortEnable is set to true, NSSF is	Subscribe with header	Status: 201 Created
	3gpp-SbiBinding: bl=nf-set;	Headers
part of nfSet.	nfset=set1.region48.amfset.5gc.mnc012.	Location: http://10.178.246.56:30075/
	mcc345	nnssf-nssaiavailability/v1/nssai-
	Helm Parameters and Values:	availability/subscriptions/1
	global.indirectCommunicationSupp ortEnable: true	<b>3gpp-Sbi-Binding:</b> bl=nf-set; nfset=set1.nssfset.5gc.mnc012.mcc345
	<pre>global.nfSet: set1.nssfset.5gc.mnc012.mcc345</pre>	Notification with headers 3gpp-Sbi-Routing-Binding: bl=nf-set;
	global.nssfApiRoot: http:// 10.178.246.56:30075/	nfset=set1.region48.amfset.5gc.mnc012. mcc345
		3gpp-Sbi-Callback: Nnssf_NSSAIAvailability_Notification
Subscription without binding,	Input message:	Subscription Response
global.indirectCommunicationSupp ortEnable is set to true, NSSF is	Subscribe without Header (Direct)	Status: 201 Created
part of nfSet.	Helm Parameters and Values:	Location: http://10.178.246.56:30075/
	global.indirectCommunicationSupp ortEnable: true	nnssf-nssaiavailability/v1/nssai- availability/subscriptions/1
	<pre>global.nfSet: set1.nssfset.5gc.mnc012.mcc345</pre>	
	global.nssfApiRoot: http:// 10.178.246.56:30075/	
Subscription with binding,	Input message:	Subscription Response
global.indirectCommunicationSupp	Subscribe with header	Status: 201 Created
ortEnable is set to false, NSSF is part of nfSet.	3gpp-Sbi-Routing-Binding: bl=nf-set; nfset=set1.region48.amfset.5gc.mnc012. mcc345	
	Helm Parameters and Values:	
	global.indirectCommunicationSupp ortEnable: false	
	global.nfSet: set1.nssfset.5gc.mnc012.mcc345	
	global.nssfApiRoot: http://	
	10.178.246.56:30075/	
Subscription without binding,	Input message:	Subscription Response
global.indirectCommunicationSupp	Subscribe without Header (Direct)	Status: 201 Created
ortEnable is set to false, NSSF is part of nfSet.	Helm Parameters and Values:	
	global.indirectCommunicationSupp ortEnable: false	
	global.nfSet:	
	set1.nssfset.5gc.mnc012.mcc345	
	global.nssfApiRoot: http:// 10.178.246.56:30075/	



Table 4-17 (Cont.) Error Scenarios

Scenario	Input Details	Output
Subscription with binding, global.indirectCommunicationSupp	Input message:	Subscription Response
	Subscribe with Header	Status: 201 Created
ortEnable is set to true, NSSF is not part of nfSet.	3gpp-Sbi-Binding: bl=nf-set;	Headers
part of filloet.	nfset=set1.region48.amfset.5gc.mnc012. mcc345	Location: http://10.178.246.56:30075/ nnssf-nssaiavailability/v1/nssai-
	Helm Parameters and Values:	availability/subscriptions/1
	<pre>global.indirectCommunicationSupp ortEnable: true</pre>	3gpp-Sbi-Routing-Binding: bl=nf-instance; nfinst=54804518-4191-46b3-955c-
	global.nfSet: Not part of nf-set but part of GR	ac631f953ed7; backupnfinst=54804518-4191-46b3-955c- ac631f953ed8
	global.nssfApiRoot: http://	
	10.178.246.56:30075/	Notification with headers
		3gpp-Sbi-Routing-Binding: bl=nf-set; nfset=set1.region48.amfset.5gc.mnc012. mcc345
		3gpp-Sbi-Callback: Nnssf_NSSAIAvailability_Notification
		ERROR Log
Subscription with binding,	Input message:	Subscription Response
global.indirectCommunicationSupp ortEnable is set to true, NSSF is not part of nfSet.	Subscribe with Header	Status 500 Internal Server Error
	3gpp-Sbi-Binding: bl=nf-set; nfset=set1.region48.amfset.5gc.mnc012. mcc345	Cause: CONFIGURATION_ERROR
	Helm Parameters and Values:	\ "type":
	global.indirectCommunicationSupp ortEnable: true	"INTERNAL_SERVER_ERROR", "title":
	global.nfSet: Not part of nf-set and not part of GR	"CONFIGURATION_ERROR", "status": 500,
	global.nssfApiRoot: http:// 10.178.246.56:30075/	<pre>"detail": "Indirect Communication is true but NFset is null and GR is also not enabled.", "instance": "null", "cause": "CONFIGURATION_ERROR" }</pre>



Table 4-17 (Cont.) Error Scenarios

Scenario	Input Details	Output
Subscription with binding,	Input message:	Subscription Response
global.indirectCommunicationSupp	Subscribe with Header	Status: 500 Internal Server Error
ortEnable is set to true, NSSF is part of nfSet.	3gpp-Sbi-Binding: bl=nf-set;	Cause: CONFIGURATION_ERROR
part of moon	nfset=set1.region48.amfset.5gc.mnc012. mcc345	
		{
	Helm Parameters and Values:	"type":
	<pre>global.indirectCommunicationSupp ortEnable: true</pre>	"INTERNAL_SERVER_ERROR",
		"title": "INVALID_LOCATION_URL",
	<pre>global.nfSet: set1.nssfset.5gc.mnc012.mcc345</pre>	"status": 500,
	global.nssfApiRoot: Invalid	"detail": "Invalid location/
	URL(Empty)	nssfApiRoot url",
	ore (Emper)	"instance": "null",
		"cause": "CONFIGURATION_ERROR"
		}
NSSF supports multiple	Input message:	Subscription Response
PLMNSubscription with binding, global.indirectCommunicationSupp ortEnable is set to true, NSSF is part of nfSet. NSSF supports PLMN	Subscribe with Header3gpp-Sbi-Binding: bl=nf-set; nfset=set1.region48.amfset.5gc.mnc100. mcc101	Status: 201 Created
		Headers
		<b>Location</b> : http://10.178.246.56:30075/
from which AMF is requesting	Helm Parameters and Values:	nnssf-nssaiavailability/v1/nssai- availability/subscriptions/1
	global.indirectCommunicationSupp ortEnable: true	3gpp-Sbi-Binding: bl=nf-set; nfset=set1.nssfset.5gc.mnc100.mcc101
	global.nfSet: set1.nssfset.5gc.mnc012.mcc345	Notification with headers
	global.nfSet: set1.nssfset.5gc.mnc100.mcc101	3gpp-Sbi-Routing-Binding: bl=nf-set; nfset=set1.region48.amfset.5gc.mnc100. mcc101
	global.nssfApiRoot: http:// 10.178.246.56:30075/	3gpp-Sbi-Callback: Nnssf_NSSAIAvailability_Notification



Table 4-17 (Cont.) Error Scenarios

Scenario	Input Details	Output
NSSF supports multiple PLMNSubscription with binding, global.indirectCommunicationSupp ortEnable is set to true, NSSF is part of nfSet. NSSF do not support PLMN from which AMF is requesting	Input message: Subscribe with Header 3gpp-Sbi-Binding: bl=nf-set; nfset=set1.region48.amfset.5gc.mnc200. mcc201  Helm Parameters and Values: global.indirectCommunicationSupp ortEnable: true global.nfSet: set1.nssfset.5gc.mnc012.mcc345 global.nfSet: set1.nssfset.5gc.mnc100.mcc101 global.nssfApiRoot: http:// 10.178.246.56:30075/	Subscription Response Status: 403 Cause: PLMN_NOT_SUPPORTED  {   "type": "FORBIDDEN",   "title": "PLMN_NOT_SUPPORTED",   "status": 403,   "detail": "Unsupported PLMN/S   received , supported plmn   list: [Plmn [mcc=100,   mnc=101], Plmn [mcc=100,   mnc=02], Plmn [mcc=310,   mnc=14], Plmn [mcc=345,   mnc=012]]",   "instance": "null",   "cause": "PLMN_NOT_SUPPORTED" }
Subscription with invalid binding header, global.indirectCommunicationSupp ortEnable is set to true, NSSF is part of nfSet.	Input message: Subscribe with Header 3gpp-Sbi-Binding: bl=nfserviceset; nfset=set1.region48.amfset.5gc.mnc012. mcc345 Helm Parameters and Values: global.indirectCommunicationSupp ortEnable: true global.nfSet: set1.nssfset.5gc.mnc012.mcc345 global.nssfApiRoot: http:// 10.178.246.56:30075/	Subscription Response Status: 400 Bad Request Cause: INVALID_INPUT_DATA  {   "type": "BAD_REQUEST",   "title": "INVALID_INPUT_DATA",   "status": 400,   "detail": "Invalid 3gpp-Sbi- Binding Header, Only following pattern is supported bl=nf-set; nfset=set <setid>.region<region id="">.amfset.5gc.mnc<mnc>.mcc<mc c="">",   "instance": "null",   "cause": "INVALID_INPUT_DATA" }</mc></mnc></region></setid>



Table 4-17 (Cont.) Error Scenarios

Scenario	Input Details	Output
Subscription with invalid binding header, global.indirectCommunicationSupp ortEnable is set to true, NSSF is part of nfSet.	Input message: Subscribe with Header3gpp-Sbi-Binding: bl=nf-set nfset=set1.amfset.5gc.mnc012.mcc345 Helm Parameters and Values: global.indirectCommunicationSupp ortEnable: true global.nfSet: set1.nssfset.5gc.mnc012.mcc345 global.nssfApiRoot: http:// 10.178.246.56:30075/	Subscription Response Status: 400 Bad Request Cause: INVALID_INPUT_DATA  {   "type": "BAD_REQUEST",   "title": "INVALID_INPUT_DATA",   "status": 400,   "detail": "Invalid 3gpp-Sbi- Binding Header, Only following pattern is   supported bl=nf-set;   nfset=set <setid>.region<region id="">.amfset.5gc.mnc<mnc>.mcc<mc c="">",   "instance": "null",   "cause": "INVALID_INPUT_DATA" }</mc></mnc></region></setid>
Subscription with invalid binding header, global.indirectCommunicationSupp ortEnable is set to true, NSSF is part of nfSet.	Input message: Subscribe with Header 3gpp-Sbi-Binding: bl=nf-set; nfset=set1.region48.amfset.5gc.mnc8120. mcc345 Helm Parameters and Values: global.indirectCommunicationSupp ortEnable: true global.nfSet: set1.nssfset.5gc.mnc012.mcc345 global.nssfApiRoot: http:// 10.178.246.56:30075/	Subscription Response Status: 400 Bad Request Cause: INVALID_INPUT_DATA  {   "type": "BAD_REQUEST",   "title": "INVALID_INPUT_DATA",   "status": 400,   "detail": "Invalid mnc 8120   in 3gpp-Sbi-Binding Header",   "instance": "null",   "cause": "INVALID_INPUT_DATA" }  Description: Invalid mnc 8120 in 3gpp-Sbi-Binding Header



Table 4-17 (Cont.) Error Scenarios

Scenario	Input Details	Output
Subscription with invalid binding header, global.indirectCommunicationSupp ortEnable is set to true, NSSF is part of nfSet.	Input message: Subscribe with Header 3gpp-Sbi-Binding: bl=nf-set; nfset=set1.region48.amfset.5gc.mnc8120. mcc345; scope=callback; scope=other-service Helm Parameters and Values: global.indirectCommunicationSupp ortEnable: true global.nfSet: set1.nssfset.5gc.mnc012.mcc345 global.nssfApiRoot: http:// 10.178.246.56:30075/	Subscription Response Status: 400 Bad Request Cause: INVALID_INPUT_DATA  {     "type": "BAD_REQUEST",     "title": "INVALID_INPUT_DATA",     "status": 400,     "detail": "Invalid 3gpp-Sbi- Binding Header, Only following pattern is     supported bl=nf-set;     nfset=set <setid>.region<region id="">.amfset.5gc.mnc<mnc>.mcc<mc c="">",     "instance": "null",     "cause": "INVALID_INPUT_DATA" }  Description: After 3 digits of MCC, there are other characters</mc></mnc></region></setid>

# Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Network Slice Selection Function Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

# 4.29 IPv6 Support

Oracle supports single stack IPv4/IPv6 addressing. IPv6 (Internet Protocol version 6) is the sixth revision of the Internet Protocol and the successor to IPv4. It functions similarly to IPv4 in that it provides the unique IP addresses necessary for Internet-enabled devices to communicate. However, it does have one significant difference, that is, it utilizes a 128-bit IP address.

# **Managing IPv6 Support**

# Enable

To enable this feature, the following prerequisites must be satisfied:

- CNE or any CNE cloud network with IPv6 (Single stack) must be enabled.
- All cluster nodes must come with IPv6 address.



## Configure

Sample configurations of IPv4 and IPv6 are given below:



Ensure NRF, DB\_Host are either in FQDN format or are in IPv6 to support IPv6.

#### IPv4

```
nrfclient:
    # Microservice level control if specific microservice need to be disabled
nrf-client:
    # This config map is for providing inputs to NRF-Client
    configmapApplicationConfig:
    profile: |-
        [appcfg]
        primaryNrfApiRoot=10.75.225.191:31515
        secondaryNrfApiRoot=10.75.225.191:31515
        nrfScheme=http
        retryAfterTime=PT120S
        nrfClientType=NSSF
        nrfClientSubscribeTypes=NSSF
```

#### IPv6

```
nrfclient:
    # Microservice level control if specific microservice need to be disabled
nrf-client:
    # This config map is for providing inputs to NRF-Client
    configmapApplicationConfig:
    profile: |-
        [appcfg]
        primaryNrfApiRoot=[fd00:10:96::aa0b]:31515
        secondaryNrfApiRoot=[fd00:10:96::95a]:31515
        nrfScheme=http
        retryAfterTime=PT120S
        nrfClientType=NSSF
        nrfClientSubscribeTypes=NSSF
```

# Observe

NSSF behavior does not change based on underlying IP layer. If the installation and services are running, the NSSF behavior remains the same.

# Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Network Slice Selection Function Troubleshooting Guide*.
- 2. Raise a service request: See My Oracle Support for more information on how to raise a service request.



# 4.30 Supports Integration with ASM

NSSF leverages the Istio or Envoy service mesh (Aspen Service Mesh) for all internal and external communication. The service mesh integration provides inter-NF communication and allows API gateway co-working with service mesh. The service mesh integration supports the services by deploying a special sidecar proxy in the environment to intercept all network communication between microservices.

See **Configuring NSSF to support ASM** in *Oracle Communication Cloud Native Core, Network Slice Selection Function Installation and Upgrade Guide* for more details on configuring ASM.

# 4.31 Supports Compression Using Accept-Encoding or Content-Encoding gzip

HTTP data is compressed before it is sent from the server, to improve transfer speed and bandwidth utilization.

**HTTP headers** let the client and the server pass additional information with an HTTP request or response.

The **Content-Encoding**, when present in response, its value indicates which encoding is applied to the entity-body. It lets the client know how to decode in order to obtain the mediatype referenced by the Content-Type header.

The **Accept-Encoding** header is used to find out the encoding supported by the server. The server responds with the type of encoding used, indicated by the Accept-Encoding response header.

# Syntax:

Accept-Encoding: gzip

Content-Encoding: gzip

#### Managing Supports Compression Using Accept-encoding/Content-encoding gzip

#### **Enable**

To enable this feature, set the value of nsavailability.contentEncodingEnabled to true in the ocnssf\_custom\_values\_25.1.100.yaml file.

#Sample to enable gzip compression

nsavailability.contentEncodingEnabled: true

#### Configure

This sample configuration shows minimum response size over which compression of response triggers, if contentEncodingEnabled is set to true.

Helm parameter maxRequestSize is the acceptable size of request.

#Sample configuration gzip compression
 # Minimum response size required for compression to happen (size is in bytes)



nsavailability.compressionMinimumResponseSize: 1024
# Maximum limit for request size
nsavailability.maxRequestSize: 1MB

#### **Observe**

The following measurements are related to *Supports compression using Accept-encoding/ Content-encoding gzip* feature:

ocnssf\_nssaiavailability\_options\_rx

ocnssf\_nssaiavailability\_options\_tx\_status\_ok

ocnssf nssaiavailability options tx status unsupportedmediatetype

For further information about Metrics and KPIs, see <u>NSSF Metrics</u> and <u>NSSF KPIs</u> sections respectively.

**Message Scenarios** 

**Table 4-18 Message Scenarios** 

Scenario	Helm Parameter (server.compression. enabled)	Response Details
AMF sends an NSAvailability PUT with Request Message size is more than max acceptable size	NA	Response code:
		413 (Request Entity Too Large error)
		Response in gzip ?:
		No
		Response Header:
		NA
Client sends HTTP OPTIONS with "Accept- encoding" of any value (blank or empty included) other than gzip	Yes	Response code:
		415 (Unsupported Media Type)
		Response in gzip ?:
		NA
		Response Header:
		Accept-Encoding: gzip
		Allowed Methods : POST, PUT, PATCH, DELETE
		Reason: Informs the client to optimize future interactions

# Maintain

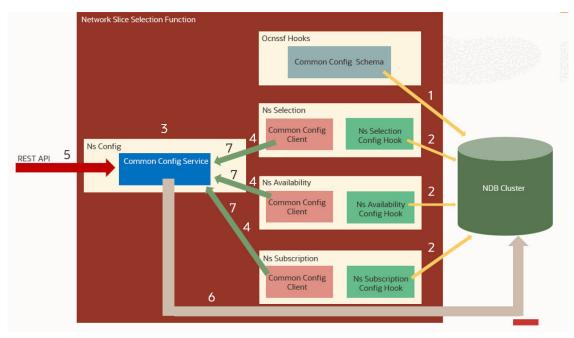
To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- **1. Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function Troubleshooting Guide*.
- 2. Raise a service request: See My Oracle Support for more information on how to raise a service request.



# 4.32 Dynamic Log Level Update

Dynamic Log Level Update allows operator to update NSSF log level dynamically without restart.



The log level can be changed by the user at run time. NSSF use common configuration service for dynamically updating Logging Information.

# **Managing Dynamic Log Level Update**

## **Enable**

- 1. Customize the ocnssf custom values 25.1.100.yaml helm file.
- 2. Set commonCfgClient.enabled to true in the helm file.

**Table 4-19 Parameters Configuration** 

Name	Default	Description
commonCfgClient.enabled	true	Enable/Disable Client.
commonCfgClient.pollingInterval	5000	Set Polling Interval in Milliseconds

# Configure

For more information on REST APIs, see **Runtime Log Level Update** in *Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide*.

## **Observe**

No new metrics or KPIs are generated for this feature. For information on other Metrics and KPIs of OCNSSF, see OCNSSF Metrics and OCNSSF KPIs sections respectively.



#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- Collect the logs: For more information on how to collect logs, see Oracle Communications
   Cloud Native Core, Network Slice Selection Function Troubleshooting Guide.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

# 4.33 NF Authentication using TLS Certificate

HTTPS support is a minimum requirement for 5G NFs as defined in 3GPP TS 33.501 Release 15. This feature enables extending identity validation from Transport layer to the Application layer and also provides a mechanism to validate the NF FQDN presence in TLS certificate as added by the Service Mesh against the NF Profile FQDN present in the request. HTTPS enables end to end encryption of messages to ensure security of data. HTTPS requires creation of TLS (Mutual TLS by 2 way exchange of ciphered keys).

#### Managing NF Authentication using TLS Certificate

## Steps to Enable HTTPS in NSSF

#### **Certificate Creation**

To create certificate user must have the following files:

- ECDSA private key and CA signed certificate of NRF (if initial algorithm is ES256)
- RSA private key and CA signed certificate of NRF (if initial algorithm is RSA256)
- TrustStore password file
- KeyStore password file
- CA certificate

#### **Secret Creation**

Execute the following command to create secret:

#### Certificate and Key Exchange

Once the connection is established, both parties can use the agreed algorithm and keys to securely send messages to each other. The handshake has 3 main phases:

- Hello
- Certificate Exchange
- Key Exchange



- 1. Hello: The handshake begins with the client sending a ClientHello message. This contains all the information the server needs in order to connect to the client via SSL, including the various cipher suites and maximum SSL version that it supports. The server responds with a ServerHello, which contains similar information required by the client, including a decision based on the client's preferences about which cipher suite and version of SSL will be used.
- 2. Certificate Exchange: Now that contact has been established, the server has to prove its identity to the client. This is achieved using its SSL certificate, which is a very tiny bit like its passport. An SSL certificate contains various pieces of data, including the name of the owner, the property (For example: domain) it is attached to, the certificate's public key, the digital signature and information about the certificate's validity dates. The client checks that it either implicitly trusts the certificate, or that it is verified and trusted by one of several Certificate Authorities (CAs) that it also implicitly trusts. The server is also allowed to require a certificate to prove the client's identity, but this only happens in very sensitive applications.
- 3. Key Exchange: The encryption of the actual message data exchanged by the client and server is done using a symmetric algorithm, the exact nature of which was agreed during the Hello phase. A symmetric algorithm uses a single key for both encryption and decryption, in contrast to asymmetric algorithms that require a public or private key pair. Both parties need to agree on this single, symmetric key, a process that is accomplished securely using asymmetric encryption and the server's public or private keys.

The client generates a random key to be used for the main, symmetric algorithm. It encrypts it using an algorithm also agreed upon during the Hello phase, and the server's public key (found on its SSL certificate). It sends this encrypted key to the server, where it is decrypted using the server's private key, and the interesting parts of the handshake are complete. The parties are identified that they are talking to the right person, and have secretly agreed on a key to symmetrically encrypt the data that they are about to send each other. HTTP requests and responses can be sent by forming a plain text message and then encrypting and sending it. The other party is the only one who knows how to decrypt this message, and so Man In The Middle Attackers are unable to read or modify any requests that they may intercept.

# NSSF supports following cipher suites

- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384
- TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384
- TLS ECDHE RSA\_WITH\_CHACHA20\_POLY1305\_SHA256
- TLS\_DHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256
- TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256

#### **HTTPS Encrypted Communication**

Once the HTTPS handshake is complete all communications between the client and the server are encrypted. This includes the full URL, data (plain text or binary), cookies and other headers.

The only part of the communication not encrypted is what domain or host the client requested a connection. This is because when the connection is initiated an HTTP request is made to the target server to create the secure connection. Once HTTPS is established the full URL is used.

This initialization only needs to occur once for each unique connection. This is why HTTP/2 has a distinct advantage over HTTP/1.1 since it multi-plexes connections instead of opening multiple connections.

# **Helm Configuration to enable HTTPS on NSSF:**



## Sample values.yaml to enable HTTPS on NSSF:

```
#Enabling it generates key and trust store for https support
    initssl: true
                        (Note: secret has to be created if its set to true)
#If true opens https port on egress gateway
  enableincominghttps: false
#Enabling it egress makes https request outside
  enableoutgoinghttps: true
 (Note: initssl should be set to true if either enableincominghttps or
enableoutgoinghttps is enabled )
#KeyStore and TrustStore related private key and Certificate configuration
(Note: The configuration names specified should be same as the
specified when creating secret)
  privateKey:
  k8SecretName: accesstoken-secret
  k8NameSpace: ocnssf
  fileName: rsa_private_key_pkcs1.pem
  certificate:
  k8SecretName: accesstoken-secret
  k8NameSpace: ocnssf
  fileName: ocnssf.cer
  caBundle:
  k8SecretName: accesstoken-secret
  k8NameSpace: ocnssf
  fileName: caroot.cer
  keyStorePassword:
  k8SecretName: accesstoken-secret
  k8NameSpace: ocnssf
  fileName: key.txt
   trustStorePassword:
  k8SecretName: accesstoken-secret
  k8NameSpace: ocnssf
  fileName: trust.txt
   initialAlgorithm: RSA256
```

# 4.34 Protection from Distributed Denial-of-Service (DDoS) Attack through Rate Limiting

Rate limiting for Ingress and Egress messages helps to prevent the DDoS attack.

NSSF uses Bucket 4j, which uses Token Bucket algorithm to enable rate limiting.

The token bucket algorithm has the following concepts:

burstCapacity: The maximum number of tokens the bucket can hold.



duration: The amount of time between the refills.

refillRate: The number of tokens that are added to the bucket during a refill.

(where duration: in seconds (M), burstCapacity: (C), refillRate: (N))

- N tokens are added to the bucket every M seconds.
- The bucket can hold at the most C tokens. If a token arrives when the bucket is full, it is discarded.

#### **Ingress Rate Limiting**

To avoid unexpected behavior and DoS attacks, NSSF allows users to enable rate limiting for ingress messages. Users can configure a cap on the maximum number of incoming messages within a given duration. Additionally, users have the option to configure a maximum cap on the number of ingress requests per service.

#### Steps to Enable Ingress Rate Limiting

NSSF allows a maximum of {burstCapacity} / {refillRate} messages within the duration specified by the parameter {duration}.

To enable ingress rate limiting at NSSF, ingress\_gateway.rateLimiting.enabled must be set to true.

#### **Global Ingress Rate Limiting**

When globalIngressRateLimiting.enabled is set to true, rate limiting is applied to all ingress messages.

#### **Route-Based Rate Limiting**

NSSF provides an option to configure route-based rate limiting and method-based rate limiting, enabling NSSF to throttle messages per service per method.

In the example below, NSSF allows 80 GET requests on the NSSelection service every 2 seconds.

Sample Ingress Rate Limiting Configuration:

```
#Rate limiting configuration
rateLimiting:
  enabled: true
routeRateLimiting:
  enabled: true
# Global rate limiting configuration
globalIngressRateLimiting:
  enabled: true
  duration: 2 # in seconds
 burstCapacity: 100
  refillRate: 1
routesConfig:
- id: nsselection_mapping
  uri: http://ocnssf-nsselection:5745
  path: /nnssf-nsselection/**
  order: 1
#Route level limiting configuration enabled for NSSelection
  methodRateLimiting: # specify the list of methods u have to rate limit
  - method: GET
    burstCapacity: 80
```



```
refillRate: 1
   duration: 2
#Route level limiting configuration not enabled for NSAvailability
- id: availability_mapping
   uri: http://ocnssf-nsavailability:5745
   path: /nnssf-nssaiavailability/**
   order: 2
- id: nsconfig_mapping
   uri: http://ocnssf-nsconfig:5755
   path: /nnssf-configuration/**
   order: 3
```

#### **Egress Rate Limiting**

NSSF sends notification messages to AMF based on configuration changes of supported SNSSAI/s in a TAI. Operators can throttle notification messages by enabling egress message rate limiting.

#### **Steps to Enable Egress Rate Limiting**

To enable rate limiting, egress-gateway.notificationRateLimit.enabled must be set to true.

In the example below, NSSF has a maximum cap of 200 notifications per second:

```
egress-gateway:
  notificationRateLimit:
    enabled: false
    duration: 1
    bucketCapacity: 200
    refillRate: 1
```

#### Observation

The following are the metrics related to Distributed Denial-of-Service (DDoS) attack prevention through rate limiting:

- oc\_ingressgateway\_global\_ratelimit
- oc ingressgateway route ratelimit
- oc\_egressgateway\_notification\_ratelimit

For further details of Metrics and KPIs, see <u>NSSF Metrics</u> and <u>NSSF KPIs</u> sections respectively.

#### Maintain

To resolve any alerts at the system or application level, see <u>NSSF Alerts</u> section. If the alerts persist, perform the following:

- 1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Network Slice Selection Function Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.



# 4.35 Automated Testing Suite Support

NSSF provides Automated Testing Suite for validating the functionalities. Through Automated Testing Suite (ATS), Oracle Communications aims at providing an end-to-end solution to its customers for deploying and testing its 5G-NFs. See *Oracle Communications Cloud Native Core, Automated Testing Suite Guide* for more information.

# Configuring NSSF using CNC Console

This chapter describes how to configure different NSSF managed objects using Oracle Communications Cloud Native Configuration Console (CNCC).

## 5.1 Support for Multicluster Deployment

CNC Console supports both single and multiple cluster deployments by facilitating NSSF deployment in local and remote Kubernetes clusters. For more information about single and multiple cluster deployments, see *Oracle Communications Cloud Native Configuration Console Installation*, *Upgrade*, and *Fault Recovery Guide*.

A single instance of CNC Console can configure multiple clusters of NSSF deployments, where each cluster has an agent console installation and a NSSF installation.

## 5.2 CNC Console Interface

This section provides an overview of the Oracle Communications Cloud Native Configuration Console (CNCC), which includes an interface to configure the NSSF features.

To configure the NSSF services using the CNCC, log in to the CNCC application. To log into CNCC, update the hosts file available at the **C:\Windows\System32\drivers\etc location** when CNCC is hosted on a third party cloud native environment.

1. In the Windows system, open the hosts file in the notepad as an administrator and append the following set of lines at the end of the hosts file:

```
<CNCC Node IP> cncc-iam-ingress-gateway.cncc.svc.cluster.local
<CNCC Node IP> cncc-core-ingress-gateway.cncc.svc.cluster.local
```

#### For example:

```
10.75.212.88 cncc-iam-ingress-gateway.cncc.svc.cluster.local 10.75.212.88 cncc-core-ingress-gateway.cncc.svc.cluster.local
```



The IP Address mentioned above may change when the deployment cluster changes.

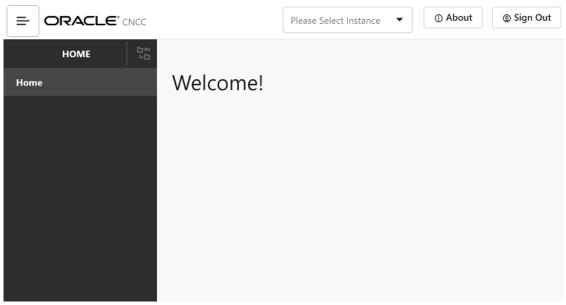
Save and close the hosts file.
 Before logging into CNC Console, create a CNCC user name and password. Log in to the
 CNC Console application using these login credentials. For information on creating a CNC
 Console user and password, see Oracle Communications Cloud Native Configuration
 Console Installation, Upgrade, and Fault Recovery Guide.



#### **CNC Console Log in**

Following is the procedure to log into CNC Console:

- Open any web browser.
- 2. Enter the URL: http://<host name>:<port number>. where, host name is cncc-iam-ingress-ip and port number is cncc-iam-ingressport.
- 3. Enter valid login credentials.
- 4. Click Log in. The CNC Console interface is displayed.



Select the required NF instance from the **Select Instance** drop-down list. The left pane displays the selected network function.

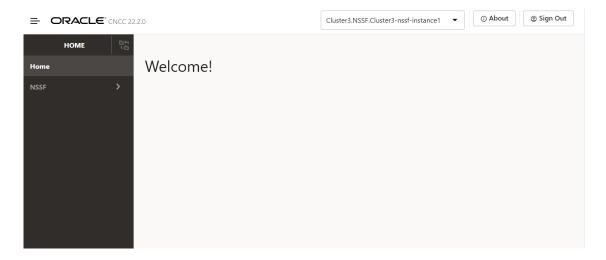
# **5.3 NSSF Configuration**

This section describes how to configure different managed objects of NSSF using CNC Console.

On selecting NSSF instance from the drop-down list, the following screen appears:



Figure 5-1 NSSF Welcome Screen



### 5.3.1 AMF Set

Perform the following procedure to configure the AMF Set:

- 1. From the left navigation menu, navigate to NSSF.
- Select NSSF and click AMF Set. The AMF Set page is displayed.
- 3. Click Add from the top right side to add AMF Resolution parameters.
- 4. Configure AMF Set fields as described in the following table:

Table 5-1 AMF Set Parameters

Field Name	Description
Region ID	Region ID of the target AMF list.
Set ID	Set ID of the target AMF list.
MCC	Mobile Country Code.
MNC	Mobile Network Code.
Salience	Order of importance (higher salience, more important). Default value is 0.
MCC-MNC-RegionID-SetID	Combination of MCC, MNC, RegionID, and SetID, separated by "-".

5. Click Save to save or Cancel to discard your progress on the Add AMF Set page.

For more information on parameter values, see Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.



Use the **Edit, Delete, View** icons available in the **Actions** column of the **AMF Set** page to update, delete, or view any preconfigured information of the AMF Set.



### 5.3.2 AMF Resolution

Perform the following procedure to configure the AMF Resolution:

- 1. In the left navigation menu, navigate to NSSF.
- 2. Select NSSF and click AMF Resolution. The AMF Resolution page is displayed.
- 3. Click **Add** from the top right side to add **AMF Resolution** parameters.
- 4. Configure AMF Resolution fields as described in the following table:

Table 5-2 AMF Resolution Parameters

Field Name	Description
Region ID	Region ID of the target AMF list.
Set ID	Set ID of the target AMF list.
MCC	Mobile Country Code.
MNC	Mobile Network Code.
MCC-MNC-RegionID-SetID	Combination of MCC, MNC, RegionID, and SetID, separated by "-".

- Click Add under Candidate AMF Lists to add the Candidate AMF Lists parameters. The Add Candidate AMF Lists pop-up window appears.
  - a. To add an AMF in the list, enter the values for Candidate AMF as described in the following table:

Table 5-3 Candidate AMF Lists Parameters

Field Name	Description
Instance ID	Instance id of the AMF

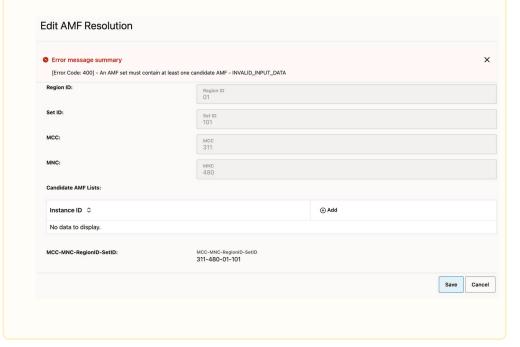
b. To delete an AMF from the list, click on the delete icon corresponding to the AMF.





An AMF set must contain at least one AMF, and this requirement must be maintained during configuration. The NSSF does not allow the operator to delete all AMFs within an AMF set, as an empty AMF set has no operational value. Therefore, if the operator intends to remove all AMFs from an AMF set, it is recommended to delete the entire AMF set instead.

Deleting all AMFs will result in an empty AMF set, triggering [Error Code: 400] when attempting to save the progress as shown in the sample image below:



- c. Click Save to save or Cancel to discard your progress in the Add Candidate AMF Lists pop-up window.
- 6. Click Save to save or Cancel to discard your progress on the Add AMF Resolution page.

#### 🛕 Caution

Discarding your progress on the **Add AMF Resolution** page will also erase any changes made in the **Add Candidate AMF Lists** pop-up. Clicking **Cancel** at this step will discard all ongoing progress (addition or deletion) on this page, including its sub-sections, such as the **Candidate AMF Lists Parameters** pop-up.

For more information on parameter values, see *Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.* 

### Note

Use the **Edit** or **View** icons available in the **Actions** column of the **AMF Resolution** page to update or view any preconfigured information of the AMF Resolution.



## 5.3.3 Configured SNSSAI

Perform the following procedure to configure SNSSAI:

- From the left navigation menu, navigate to **NSSF**.
- Select NSSF and click Default Configured SNSSAI. The **Configured SNSSAI** page is displayed.
- Click **Add** from the top right side to add **Configured SNSSAI** parameters.
- Configure **Configured SNSSAI** fields as described in the following table:

Table 5-4 Configured SNSSAI Parameters

Field Name	Description
MCC	Mobile Country Code.
MNC	Mobile Network Code.
PLMN ID	Combination of MCC and MNC, separated by "-".

- 5. Click Add under NSSAI to add the NSSAI parameters. The Add NSSAI pop-up window appears.
- Enter the values for **Add NSSAI** parameters as described in the following table:

Table 5-5 NSSAI parameters

Field Name	Description
SST	Slice or Service Type.
SD	Slice Differentiator.

- Click Save to save or Cancel to discard your NSSAI configuration in the Add NSSAI popup window.
- Click Save to save or Cancel to discard your progress on the Add Configured SNSSAI page.

For more information on parameter values, see Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.



#### (i) Note

Use the Edit, Delete, View icons available in the Actions column of the Configured **SNSSAI** page to update, delete, or view any preconfigured information of the Configured SNSSAI.

## 5.3.4 Georedundant Sites

Perform the following procedure to configure Georedundant Sites:

- From the left navigation menu, navigate to **NSSF**.
- Select NSSF and click Georedundant Sites. The **Georedundant Sites** page is displayed.



- 3. Click **Add** from the top right side to add **Georedundant Sites** parameters.
- 4. Configure **Georedundant Sites** fields as described in the following table:

**Table 5-6 Georedundant Sites Parameters** 

Field Name	Description
NF ID	Instance ID of the NSSF site.
Rank	The priority given by the operator to related georedundant sites.
Georedundant Site Status	Current status of the site or NSSF, status can be either ACTIVE or DOWN.

Click Save to save or Cancel to discard your progress on the Add Georedundant Site page.

For more information on parameter values, see *Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.* 



Use the **Edit, Delete, View** icons available in the **Actions** column of the **Georedundant Sites** page to update, delete, or view any preconfigured information of the Georedundant Sites.

### 5.3.5 NSI Profile

Perform the following procedure to configure NSI Profile:

- 1. From the left navigation menu, navigate to NSSF.
- Select NSSF and click NSI Profile. The NSI Profile page is displayed.
- 3. Click **Add** from the top right side to add **NSI Profile** parameters.
- 4. Configure **NSI Profile** fields as described in the following table:

Table 5-7 NSI Profile Parameters

Field Name	Description
Name	Network Slice Instance Profile Name.
NRF URI	URI of the Network Repository Function.
NRF NF Management URI	Management URI of Network Resource Function.
NRF Access Token URI	Access Token URI of Network Resource Function.
Network Slice Instance Identifier	Network Slice Instance Identifier code.
MCC	Mobile Country Code.
MNC	Mobile Network Code.

- Click Add under Target AMF Sets to add the Target AMF Set parameters. The Add Target AMF Sets pop-up window appears.
- 6. Enter the values for Add Target AMF Sets parameters as described in the following table:



**Table 5-8 Target AMF Sets parameters** 

Field Name	Description
Region ID	Region ID of Target AMF Set
Set ID	Set ID of Target AMF Set.
Salience	Salience of Target AMF Set.

- 7. Click **Save** to save or **Cancel** to discard your Target AMF Set configuration in the **Add** Target AMF Set pop-up window.
- 8. Click Save to save or Cancel to discard your progress on the Add NSI Rule Profile page.



#### Note

Use the Edit, Delete, View icons available in the Actions column of the NSI Profile page to update, delete, or view any preconfigured information of the NSI Profile.

### 5.3.6 NSSAI Auth

The configuration of NSSAI Auth denotes the mapping of allowed and restricted SNSSAI per TAI. It enables the configuration of network slice authentication rules by configuring Grant status (Allowed\_PLMN, Rejected\_PLMN, or Rejected\_TAC) for S-NSSAI on a per TAI basis.

#### **Query Parameters**

Use Query Parameters to send GET or DELETE requests for the specified NSSAI Auth by providing the value of the "name" parameter of a target configured Network Slice Authentication Rule.

#### For example:

If a Network Slice Authentication Rule Name is "2-AUTH-null-100001-200010-2-EABB02", enter this value in the **Ouery Parameters** and Click **Get** or **Delete** respectively to fetch details of this rule or remove this configured Network Slice Authentication Rule.



#### (i) Note

If the response data is above the configured display limit, a message is displayed stating "Response data has crossed the configured display limit (5 MB), please click on **Export** to download it as a file". Currently, the display limit cannot be modified, it is set to 5 MB.

To know more about the "name" parameter, see the table below or *Oracle Communications* Cloud Native Core, Network Slice Selection Function REST Specification Guide.

#### **Configuring NSSAI Auth**

Perform the following procedure to configure NSSAI Auth:

- 1. From the left navigation menu, navigate to NSSF.
- Select NSSF and click NSSAI Auth.



The **NSSAI Auth** page is displayed.

- Click Add from the top right side.A tabbed interface for Response and Request body appears.
- 4. Click on **Request** tab to configure request body parameters in JSON format.
- **5.** Configure request body using the parameters described in the following table:

Table 5-9 NSSAI Auth Parameters

Field Name	Description
name	Network Slice Authentication Rule Name.
plmnId	Public Land Mobile Network ID (MCC:MNC).
tac	Tracking Area Code.
tacrange	Range of TAC represented by starttac and endtac. Either tac or tacrange would be present. If both are not present, then Auth corresponds to PLMN.
starttac	A 4/6 digit hexadecimal number that identifies starting value of a Tracking Area in a TAC range.
endtac	A 4/6 digit hexadecimal number that identifies ending value of a Tracking Area in a TAC range.
snssai	Single Network Slice Selection Assistance Information.
sst	Slice or Service Type
sd	Slice Differentiator
grant	Whether the requested s-NSSAI is 'ALLOWED' or 'RESTRICTED'.

- 6. Click **Submit** to send request with the configured request body parameters.
- Click Response to see the response body of the sent request.

For more information on parameter values, see *Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.* 

#### (i) Note

- Use the Edit and Delete options on the NSSAI Auth page to update or delete a request.
- Use Get option without providing the Query Parameters to view all existing Requests and corresponding Responses.
- Use Export option to download the response data as a JSON file.
- Use Clear option to clear the Request and Response panes.

## **5.3.7 NSS Rule**

The NSS Rules Managed Object enables the configuration of policy rules. It enables an operator to allow, reject, or associate a Network slice based on NSSAI (SST and SD), PLMN (MCC and MNC), TAC, and AMF\_ID. The operator can configure the salience value to prioritize rules. A higher salience value implies a higher priority of the rule.



#### **Query Parameters**

Use **Query Parameters** to send GET or DELETE requests for the specified NSS Rule by providing the value of the "name" parameter of a target configured Network Slice Selection Rule.

#### For example:

If a Network Slice Selection Rule Name is "TACRANGE-SNSSAI-1-RULE-4", then enter this value in the **Query Parameters** and Click **Get** or **Delete** respectively to fetch details of this rule or remove this configured Network Slice Selection Rule.

#### Note

If the response data is above the configured display limit, a message is displayed stating "Response data has crossed the configured display limit (5 MB), please click on **Export** to download it as a file". Currently, the display limit cannot be modified, it is set to 5 MB.

To know more about the "name" parameter, see the table below or *Oracle Communications Cloud Native Core*, *Network Slice Selection Function REST Specification Guide*.

#### **Configuring NSS Rule**

Perform the following procedure to configure NSS Rule:

- 1. From the left navigation menu, navigate to NSSF.
- 2. Select NSSF and click NSS Rule. The NSS Rule page is displayed.
- Click Add from the top right side.A tabbed interface for Response and Request body appears.
- 4. Click on **Request** tab to configure request body parameters in JSON format.
- **5.** Configure request body using the parameters described in the following table:

Table 5-10 NSS Rule Parameters

Field Name	Description
name	Network Slice Selection Rule Name.
amfId	AMF Identifier.
plmnId	Public Land Mobile Network ID (MCC:MNC).
tac	Tracking Area Code.
tacrange	Range of TAC represented by starttac and endtac. Either tac or tacrange would be present. If both are not present, then Auth corresponds to PLMN.
starttac	A 4/6 digit hexadecimal number that identifies starting value of a Tracking Area in a TAC range.
endtac	A 4/6 digit hexadecimal number that identifies ending value of a Tracking Area in a TAC range.
snssai	Single Network Slice Selection Assistance Information.
sst	Slice or Service Type



Table 5-10 (Cont.) NSS Rule Parameters

Field Name	Description
sd	Slice Differentiator
salience	The order of importance (higher salience, more important).
behavior	Behavior of the parameter.
accessType	"3GPP_ACCESS" or "NON_3GPP_ACCESS"
nsiProfiles	An array of NsiProfile map, which contains name and salience of the NSI Profile.

- 6. Click **Submit** to send request with the configured request body parameters.
- 7. Click **Response** to see the response body of the sent request.

#### Note

- Use the Edit and Delete options on the NSS Rule page to update or delete a request.
- Use Get option without providing the Query Parameters to view all existing Requests and corresponding Responses.
- Use Export option to download the response data as a JSON file.
- Use Clear option to clear the Request and Response panes.

## 5.3.8 Time Profile

Perform the following procedure to configure Time Profile:

- 1. From the left navigation menu, navigate to NSSF.
- 2. Select **NSSF** and click **Time Profile**. The **Time Profile** page is displayed.
- 3. Click **Add** from the top right side to add **Time Profile** parameters.
- 4. Configure **Time Profile** fields as described in the following table:

**Table 5-11** Time Profile Parameters

Field Name	Description
Name	Time Profile Name.
Start Date	Date in the format of yy-mm-dd.
End Date	Date in the format of yy-mm-dd.
Days Of Week	Name of the day.

- 5. Click **Add** under **Time Spans** to add the Time Spans parameters. The **Add Time Spans** pop-up window appears.
- 6. Enter the values for Add Time Spans parameters as described in the following table:



Table 5-12 Time Span Parameters

Field Name	Description	
Start Time	Start time in hh:mm:ss format.	
End Time Date	End time in hh:mm:ss format.	

- Click Save to save or Cancel to discard your NSSAI configuration in the Add Time Spans pop-up window.
- 8. Click **Save** to save or **Cancel** to discard your progress on the **Add Time Profile** page.



Use the **Edit, Delete, View** icons available in the **Actions** column of the **Time Profile** page to update, delete, or view any preconfigured information of the Time Profile.

## 5.3.9 Logging Level Options

Perform the following procedure to configure Logging Level Options:

- From the left navigation menu, navigate to NSSF.
- Select NSSF and click Logging Level Options. The Logging Level Options page is displayed.

This page displays a list of preconfigured log levels with the following details:

- Service
- Application Log Level
- 3. Click **View** on the right most column of a log level from the list to see the preconfigured log level details in a pop-up window named **View Log Level List**.
- Click X icon to close View Log Level List pop-up window.
- 5. Click **Edit** from the top right side to edit **Logging Level Options** parameters.
- 6. Configure **Logging Level Options** fields as described in the following table:

Table 5-13 Logging Level Options Parameters

Field Name	Description
Service Type	Select the service type you want to configure from the drop-down list with the following options:  nsavailability nsselection nsaudit nsconfig nssubscription egw igw nrf-client-nfdiscovery nrf-client-nfmanagement



Table 5-13 (Cont.) Logging Level Options Parameters

Field Name	Description
Application Log Level	Select log level for the application from the drop-down list with the following options:  DEBUG  ERROR  INFO  TRACE  WARN  FATAL
Package Log Level	This is a list of packages with corresponding log level applicable to the selected Service Type.

- Click Edit under Package Log Level to edit the Package Log Level parameters for the selected Service Type. The Edit Package Log Level pop-up window appears.
- 8. Enter the values for Edit Package Log Level parameters as described in the following table:

Table 5-14 Package Log Level Parameters

Field Name	Description
Package	This field is non editable. It is preconfigured based on the selected Service Type.
Log Level	Select log level for the package from the drop-down list with the following options:  DEBUG  ERROR  INFO  TRACE  WARN  FATAL

- Click Save to save or Cancel to discard your progress in the Edit Package Log Level pop-up window.
- Click Save to save or Cancel to discard your progress on the Edit Logging Level
   Options page.

## 5.3.10 PLMN Level NSI Profiles

Perform the following procedure to configure PLMN Level NSI Profiles:

- 1. In the left navigation menu, navigate to NSSF.
- Select NSSF and click PLMN Level NSI Profiles. The PLMN Level NSI Profiles page is displayed.
- 3. Click Add from the top right side to add PLMN Level NSI Profiles parameters.
- 4. Configure PLMN Level NSI Profiles fields as described in the following table:



Table 5-15 PLMN Level NSI Profiles Parameters

Field Name	Description
Name	Name of the PLMN Level NSI Profile.
NRF URI	URI of the Network Repository Function.
NRF NF Management URI	Management URI of Network Resource Function.
NRF NF Access Token URI	Access Token URI of Network Resource Function.
Network Slice Instance Identifier	Network Slice Instance Identifier (nsid).
MCC	Mobile Country Code.
MNC	Mobile Network Code.
PLMN ID	Combination of MCC, MNC, RegionID, and SetID, separated by "-".

Click Save to save or Cancel to discard your progress on the Add PLMN Level NSI Profile page.

For more information on parameter values, see Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.



#### (i) Note

Use the Edit, Delete, View icons available in the Actions column of the PLMN Level NSI Profile page to update, delete, or view any preconfigured information of the PLMN Level NSI Profile.

## 5.3.11 Mapping of Nssai

Mapping of Nssai is a mandatory Managed Object to support request mapping. It is added to support EPS to 5G handover and contains mapping of 4G S-NSSAI to 5G S-NSSAI for a given PLMN.

This Managed Object must be configured for each supported PLMN.

#### **Query Parameters**

Use Query Parameters to send GET, UPDATE, or DELETE requests for the specified Mapping of Nssai by providing the value of the "mcc" and "mnc" parameters is the following format.

mcc=<value>&mnc=<value>

#### For example:

If mcc and mnc are 100 and 101 respectively, enter these values in the Query Parameters and click Get, Edit, or Delete respectively to fetch, Update, or Delete details.

To know more about the parameters, see Table 5-16 or Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.

#### **Configuring Mapping of Nssai**

Perform the following procedure to configure Mapping of Nssai:

- From the left navigation menu, navigate to **NSSF**.
- Select **NSSF** and click **Mapping of Nssai**.



The **Mapping of Nssai** page is displayed.

- Click Add from the top right side.A tabbed interface for Response and Request body appears.
- 4. Click on **Request** tab to configure request body parameters in JSON format.
- 5. Configure request body using the parameters described in the following table:

Table 5-16 Mapping of Nssai Parameters

Field Name	Description
mcc	Specifies Mobile Country Code.
mnc	Specifies Mobile Network Code.
mappingOfNssai	Specifies an array of MappingOfSnssai. For more information, see <u>Table 5-17</u> .

#### Table 5-17 MappingOfSnssai

Attribute	Description
servingSnssai	This IE specifies the S-NSSAI value of serving network (5G).
homeSnssai	This IE specifies the mapped S-NSSAI value of home network (4G).

- 6. Click **Submit** to send request with the configured request body parameters.
- 7. Click **Response** to see the response body of the sent request.

For more information on parameter values, see *Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.* 

#### (i) Note

- Use the Edit and Delete options on the Mapping of Nssai page to update or delete a request.
- Use Get option without providing the Query Parameters to view all existing requests and corresponding responses.
- Use Export option to download the response data as a JSON file.
- Use Clear option to clear the request and response panes.

## 5.3.12 NSSF Restore

This API provides the functionality to restore an existing configuration restored as backup.

To know more about the parameters, see <u>Table 5-18</u> or Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.

#### **Configuring NSSF Restore**

Perform the following procedure to configure NSSF Restore:

- 1. From the left navigation menu, navigate to NSSF.
- 2. Select NSSF and click NSSF Restore.



The **NSSF Restore** page is displayed.

- Click Add from the top right side.A tabbed interface for Response and Request body appears.
- 4. Click on **Request** tab to configure request body parameters in JSON format.
- 5. Configure request body using the parameters described in the following table:

**Table 5-18 NSSF Restore Parameters** 

Field Name	Description
NsiProfile	If NsiProfile is configured, this parameter contains list of NsiProfile.
NssaiAuth	If NssaiAuth is configured, this parameter contains list of NssaiAuth.
TimeProfile	If TimeProfile is configured, this parameter contains list of TimeProfile.
NssRule	If NssRule is configured, this parameter contains list of NssRule.
AmfResolution	If AmfResolution is configured, this parameter contains list of AmfResolution.
ConfiguredSnssai	If ConfiguredSnssai is configured, this parameter contains list of ConfiguredSnssai.

- 6. Click **Submit** to send request with the configured request body parameters.
- 7. Click **Response** to see the response body of the sent request.

For more information on parameter values, see *Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.* 

### Note

- Use Export option to download the response data as a JSON file.
- Use Clear option to clear the Request and Response panes.

## 5.3.13 NSSF Backup

This API provides the functionality to backup an existing configuration.

#### **Configuring NSSF Backup**

Perform the following procedure to configure NSSF Backup:

- 1. From the left navigation menu, navigate to NSSF.
- Select NSSF and click NSSF Backup. The NSSF Backup page is displayed.
- Click Get from the top right side.
   A panel interface for Response appears, which contains the response body of the sent Get request.

For more information on REST API, see *Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.* 



- Use Export option to download the response data as a JSON file.
- Use Clear option to clear the Request and Response panes.

## 5.3.14 NSSF System Option

- From the left navigation menu, navigate to NSSF and click NSSF System Option.
   The NSSF System Option page is displayed.
- Click Add to add NSSF System Option parameters.A tabbed interface for Response and Request body appears.
- 3. Click on **Request** tab to configure request body parameters in JSON format.
- 4. Configure request body using the parameters described in the following table:

**Table 5-19 NSSF System Option Parameters** 

Parameter	Description
NssfSystemOptionService	It is the main object that contains other objects required for enhanced computation of allowedNSSAI in NSSF.
NssfSystemOptionService.scope	Name of the NSSF Service that the feature applies to. Currently, it is limited to NsSelection only.
NssfSystemOptionService.PerPLMNConfig uration	It is the object that contains objects for PLMN level configurations required for enhanced computation of allowedNSSAI in NSSF.
NssfSystemOptionService.PerPLMNConfig uration.PlmnConfiguration	It contains objects for PLMN level configurations required for enhanced computation of allowedNSSAI in NSSF.
NssfSystemOptionService.PerPLMNConfig uration.PlmnConfiguration.EnableEnhan cedAllowedNSSAIComputation	Parameter to enable or disable the feature for a PLMN
NssfSystemOptionService.PerPLMNConfig uration.plmnId	Object that contains PLMN details
NssfSystemOptionService.PerPLMNConfig uration.plmnId.mnc	MNC of the PLMN
NssfSystemOptionService.PerPLMNConfig uration.plmnId.mcc	MCC of the PLMN



Table 5-19 (Cont.) NSSF System Option Parameters

Parameter	Description
AutoConfigurationFromNsAvailability	This parameter is used to configure NSSF's behavior with respect to "Auto-Population of Configuration Based on NSAvailability Update" feature.
	It is an enum with the following possible values:
	<ul> <li>Disable: This signifies the feature is disabled and is the default value. There will be no configuration update based on NsAvailability data if the value is disabled.</li> <li>FromTrustedAMFs: This signifies NsAvailability Update only from trusted AMFs may lead to an update in NSSF</li> </ul>
	configuration.
	FromAllAMFs: This signifies that     NsAvailability Update from any AMF may lead to an update in NSSF configuration.
	<b>Note</b> : Assuming that NsAvailability Update is processed by NSSF. That is, the AMF is authorized to send the availability update.
EnhancedPatchBehaviour	This parameter is used to allow patch remove request to remove last element of supportedSnssaiList and update AMF with empty list in database.
	If flag value is true, then we should allow the patch remove request to remove last element of supportedSnssaiList.

- 5. Click **Submit** to send request with the configured request body parameters.
- 6. Click **Response** to see the response body of the sent request

#### Note

- Use Get option without providing the Query Parameters to view all existing Requests and corresponding Responses.
- Use Export option to download the response data as a JSON file.
- Use Clear option to clear the Request and Response panes.

## 5.3.15 Trusted Amf

- From the left navigation menu, navigate to NSSF and click Trusted Amf.
  The Trusted Amf page is displayed.
- Click Add to add Trusted Amf parameters.A tabbed interface for Response and Request body appears.
- 3. Click on **Request** tab to configure request body parameters in JSON format.
- 4. Configure request body using the parameters described in the following table:



Table 5-20 Request or Response Body Parameters

Parameter	Description
amfSet	It contains the AMF Set.
amfIdList	It contains a list of AMF Identifiers.

- Click Submit to send request with the configured request body parameters.
- 6. Click **Response** to see the response body of the sent request

#### (i) Note

- Use **Get** option without providing the Query Parameters to view all existing Requests and corresponding Responses.
- Use **Edit** option to edit the request data in the Request pane.
- Use Export option to download the response data as a JSON file.
- Use Clear option to clear the Request and Response panes.

## 5.4 Common Services Configuration

Use this section to navigate to respective sections of **Egress Gateway** and **Ingress Gateway** configurations.

## 5.4.1 Egress Gateway

This section contains the Egress Gateway APIs.

## 5.4.1.1 Peer Configuration

This URI is used to add or update the list of peers wherein each peer consists of ID, host, port or virtualHost, and apiPrefix. The ID of each peer is mapped to Peer Identifier in Peer Set Configuration. The default value is null.

Perform the following configurations:

- From the left navigation menu, navigate to NSSF and click the Common Configuration option.
- 2. Click the **Egress Gateway** option to configure the Egress Gateway APIs.
- Click the Peer Configuration option under Egress Gateway to configure peers.The Peer Configuration page is displayed.
- 4. Click **Add** to add the peer configuration.

The Create Peer Configuration page is displayed.

- 5. Configure the following fields in the **Create Peer Configuration** page:
  - a. ID: Enter a unique Peer identifier.
  - b. Host: Enter the Host details of a local peer. It can be IPv4, IPv6, and FQDN details.



- c. **Port**: Enter the port details of the local host peer.
- d. API Prefix: Enter the API prefix details of a peer. Note: It is recommended to set the value as /.
- Virtual Host: Enter the Host details of a remote peer. This FQDN is sent to an alternate route service.
- f. healthApiPath: Parameter to support SCP health check API. It contains path of the health API.

#### Note

The value of this parameter should be configured to align with the SCP configuration.

6. Click **Save** on the **Create Peer Configuration** page to save the details. Click **Cancel** to discard your progress and go back to **Peer Configuration** page.

#### Note

- Use Edit icon available in the next column of the specific entry to update configured the Peer Configuration information.
- Use Refresh icon to refresh the list of peers configured.

For more information on recommended parameter values, range, default values, and whether they are mandatory, optional, or conditional, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function REST Specification Guide*.

## 5.4.1.2 Peer Set Configuration

This URI is used to add or update the list of peer sets wherein each peer set consists of id and list of http/https instances. Each instance consists of priority and peer identifier that is mapped to id in peerconfiguration resource. The id of each peer set is mapped to peerSetIdentifier in routesconfiguration resource. The default value is null.

- From the left navigation menu, navigate to NSSF and click the Common Configuration option.
- 2. Click the **Egress Gateway** option to configure the Egress Gateway APIs.
- 3. Click the **Peer Set Configuration** option under **Egress Gateway** to configure peers. The **Peer Set Configuration** page is displayed.
- Click Add to add the peer set configuration.
   The Create Peer Set Configuration page is displayed.
- 5. Configure the following fields in the **Create Peer Configuration** page:
  - a. ID: Enter a unique Peer set identifier.
  - Click Add in HTTP Configuration section to add HTTP Configuration. The Add HTTP Configuration dialog box is displayed.
    - i. Enter the following information on this page:



- . **Priority**: Enter the Priority of peer to be used in a peer set.
- ii. **Peer Identifier**: Enter the Peer identifier is the value of peer configured during PeerConfiguration.
- ii. Click **Save** to save HTTP Configuration. Click **Cancel** to discard your progress, close the dialog box, and go back to **Create Peer Set Configuration** page.
- c. Click Add in HTTPS Configuration section to add HTTPs Configuration. The Add HTTPS Configuration page is displayed.
  - i. Enter the following information on this page:
    - i. **Priority**: Enter the Priority of peer to be used in a peer set.
    - ii. Peer Identifier: Enter the Peer identifier is the value of peer configured during Peer Configuration.
  - ii. Click **Save** to save HTTP Configuration. Click **Cancel** to discard your progress, close the dialog box, and go back to **Create Peer Set Configuration** page.
- 6. Click **Save** on the **Create Peer Set Configuration** page to save the details. Click **Cancel** to discard your progress and go back to **Peer Set Configuration** page.

- Use Edit icon available in the next column of the specific entry to update configured the Peer Set Configuration information.
- Use Refresh icon to refresh the list of peer sets configured.

For more information on recommended parameter values, range, default values, and whether they are mandatory, optional, or conditional, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function REST Specification Guide*.

## 5.4.1.3 Peer Monitoring Configuration

This URI is used to update the peer configuration with healthApiPath.

Perform the following configurations:

- 1. From the left navigation menu, navigate to **NSSF** and click the **Common Configuration** option.
- 2. Click the **Egress Gateway** option to configure the Egress Gateway APIs.
- Click the Peer Monitoring Configuration option under Egress Gateway to configure peers.

The **Peer Monitoring Configuration** page is displayed with default configured values.

- Click Edit to update the peer monitoring configuration.
   The Edit Peer Monitoring Configuration page is displayed.
- 5. Configure the following fields in the **Edit Peer Monitoring Configuration** page:
  - **a. Enabled**: Use the switch to enable or disable peer monitoring feature.
  - **b. Timeout**: Attribute to configure the duration of time after which calls to the SCP health API is timed out.
  - c. Frequency: Indicates the frequency or recurring interval at which Egress Gateway initiates health check calls toward SCP.



- d. **FailureThreshold**: Indicates the number of failure responses after which a healthy SCP can be marked as unhealthy.
- e. **SuccessThreshold**: It indicates the number of successful responses after which an unhealthy SCP can be marked as healthy.
- 6. Click Save on the Edit Peer Monitoring Configuration page to save the details. Click Cancel to discard your progress and go back to Peer Monitoring Configuration page.

Use Refresh icon to refresh the peer monitoring configuration.

For more information on recommended parameter values, range, default values, and whether they are mandatory, optional, or conditional, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function REST Specification Guide*.

## 5.4.1.4 Routes Configuration

This URI is used to add or update list of routes.

Perform the following configurations:

- From the left navigation menu, navigate to NSSF and click the Common Configuration option.
- 2. Click the **Egress Gateway** option to configure the Egress Gateway APIs.
- Click the Routes Configuration option under Egress Gateway to configure peers. The Routes Configuration page is displayed.
- Click Add to add the peer configuration.
   The Create Routes Configuration page is displayed.
- 5. Configure the following fields in the **Create Routes Configuration** page:
  - a. ID: Enter a unique route configuration identifier.
  - **b. URI**: Provide any dummy URL, or leave the existing URL with existing value.
  - c. Order: Provide the order of the execution of this route.
  - d. Configure the following fields in the **metadata** section:
    - i. **httpsTargetOnly**: Enable it to select SBI instances for https list only (if 3gpp sbi target root header is http). Keep it disabled to select as per provided scheme.



SBI Routing feature will not work if this switch is disabled.

ii. httpRuriOnly: This switch indicates the scheme of the outgoing request from NSSF. If it is enabled, the scheme of RURI is changed to http. If it is disabled, no change occurs to the scheme.





SBI Routing feature will not work if this switch is disabled.

- iii. sbiRoutingEnabled: Switch to enable or disable SBI Routing feature.
- predicates: Click Add in the right side column to add predicates.
   Add predicates window is displayed.
- f. Configure the following fields in the **Add predicates** window:
  - i. pattern: Enter pattern details.
  - ii. Name: Enter name of the predicate.
- g. Click Save to save predicates configuration. Click Cancel to discard your progress, close the window, and go back to Create Routes Configuration page.
- Filters: Click Add in the right side column to add filters.
   Add Filters window is displayed.
- i. Configure the following fields in the Add Filters window:
  - i. PeerSetIdentifier: Enter PeerSetIdentifier for the filter.
  - customPeerSelectorEnabled: Use this switch to enable or disable Custom Peer Selector.
  - iii. **errorHandling**: Click **Add** in the right side column to add errorHandling scenarios. **Add errorHandling** window is displayed.
    - i. Configure the following fields in the Add errorHandling window:
      - errorCriteriaSet: Enter errorCriteriaSet.
      - actionSet: Enter actionSet.
      - iii. priority: Enter priority.
    - Click Save to save errorHandling configuration. Click Cancel to discard your progress, close the window, and go back to Add Filters window.
- i. Click **Add** at the bottom of the window to save filters configuration.
- k. Click Remove to remove filter configuration.Add Filters window is displayed with reset fields.
- Click Save in Add Filters window to save your progress (Add Filter or Remove Filter).
   Click Cancel to discard your progress, close the window, and go back to Create Routes Configuration page.
- Click Save on the Create Routes Configuration page to save the details. Click Cancel to discard your progress and go back to Routes Configuration page.

#### Note

- Use Edit icon available in the next column of the specific entry to update configured information.
- Use Refresh icon to refresh the routes configuration.



For more information on recommended parameter values, range, default values, and whether they are mandatory, optional, or conditional, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function REST Specification Guide*.

### 5.4.1.5 SBI Error Action Sets

This URI is used to list or update SBI error action sets configuration at Egress gateway. By default this configuration is disabled.

Perform the following configurations:

- From the left navigation menu, navigate to NSSF and click the Common Configuration option.
- Click the Egress Gateway option to configure the Egress Gateway APIs.
- Click the SBI Error Action Sets option under Egress Gateway to configure peers. The SBI Error Action Sets page is displayed.
- Click Add to add an error action set configuration.
   The Create SBI Error Action Sets page is displayed.
- Configure the following fields in the Create SBI Error Action Sets page:
  - a. ID: Enter an unique ID for SBI routing error action set.
  - Action: Indicate the action that needs to be taken when specific criteria set is matched.
  - **c. Attempts**: Enter the maximum number of retries to either same or different peer in case of error or failures from backend.
  - d. Block List: Perform the following configuration:
    - i. **Enabled**: Use the switch to disable or enable the peer bloacking feature using the server headers received in the response.
    - **ii. Duration**: Enter the duration for which the peer is blocked and no traffic is routed to that peer for this period.
- Click Save on the Create Routes Configuration page to save the details. Click Cancel to discard your progress and go back to Create Routes Configuration page.

#### Note:

- Use Edit icon available in the next column of the specific entry to update the Error Action Sets information.
- Use Refresh icon to refresh the configuration.

### 5.4.1.6 SBI Error Criteria Sets

This URI is used to list or update SBI error criteria sets configuration at Egress Gateway. By default, this configuration is disabled.

- From the left navigation menu, navigate to NSSF and click the Common Configuration option.
- Click the Egress Gateway option to configure the Egress Gateway APIs.
- Click the SBI Error Criteria Sets option under Egress Gateway to configure peers.The SBI Error Criteria Sets page is displayed.



- Click Add to add an error action set configuration.
   The Create SBI Error Criteria Sets page is displayed.
- 5. Configure the following fields in the Create SBI Error Criteria Sets page:
  - a. ID: Enter an unique ID for SBI routing error action set.
  - b. Method: Indicate the type of methods for which the re-route need to be attempted.
  - c. Exceptions: Enter the specific exceptions for which reroute or retry will be triggered.
  - d. Response: Configure the following fields under Statuses section. Click Add to add HTTP status details:
    - Status Series: Enter the HTTP status series for which reroute or retry is triggered, when the error response is received from downstream.
    - ii. Status: Specify HTTP statuses that belongs to above mentioned status series for which reroute or retry is triggered. To enable retry or reroute for all the HTTP status belonging to a status series, configure this as -1.
    - iii. Click Save in Add Statuses window to save statuses configuration. Click Cancel to discard your progress, close the window, and go back to Create SBI Error Criteria Sets page.
- Click Save on the Create SBI Error Criteria Sets page to save the details. Click Cancel to discard your progress and go back to Create SBI Error Criteria Sets page.

#### Note:

- Use Edit icon available in the next column of the specific entry to update the SBI Error Criteria Sets information.
- Use Refresh icon to refresh the configuration.

For more information about the configuration parameters, see *Oracle Communication Cloud Native Core*, *Network Repository Function REST Specification Guide*.

## 5.4.1.7 User Agent Header Generation

This URI is used to Enable or Disable User-Agent Header.

- From the left navigation menu, navigate to NSSF and click the Common Configuration option.
- 2. Click the **Egress Gateway** option to configure the Egress Gateway APIs.
- Click the User Agent Header Generation option under Egress Gateway to configure peers.
  - The **User Agent Header Generation** page is displayed with default configured values.
- Click Edit to update the User Agent Header Generation.
   The Edit User Agent Header Generation page is displayed.
- 5. Configure the following fields in the Edit User Agent Header Generation page:
  - a. **Enabled**: Use the switch to enable or disable User Agent Header feature.
  - **b. NF Type:** : Attribute to configure the nfType that is used to generate the User Agent Header. In this case, it is NSSF.
  - c. NF Instance ID: : Indicates the UUID (Instance ID) of the NSSF deployment used to generate the User Agent Header.



- d. NF FQDN: : This is an optional parameter, if operators want to include the FQDN string configured under this section then the parameter Add Fqdn To Header needs to be enabled.
- **e.** Add Fqdn To Header: : Use the allow or deny User Agent from appending the NSSF FQDN information while generating the User Agent Header.
- f. Overwrite Header: Use this switch to govern if you want to include the User Agent Header generated at NSSF Egress Gateway or forward the User Agent received from service request.

- When User Agent Header is enabled but the header information is missing, then it is picked from the OAuthClient module.
- If the User Agent Header is present in the request towards AMF or NRF, then the value present in the header is overwritten or forwarded based on the Overwrite Header switch. If this switch is enabled, then the header is overwritten.
- 6. Click **Save** on the **Edit User Agent Header Generation** page to save the details. Click **Cancel** to discard your progress and go back to **User Agent Header Generation** page.

#### Note

• Use **Refresh** icon to refresh the User Agent Header Generation.

For more information on recommended parameter values, range, default values, and whether they are mandatory, optional, or conditional, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function REST Specification Guide*.

## 5.4.2 Ingress Gateway Configuration

This section contains the Ingress Gateway APIs.

### 5.4.2.1 Error Code Profiles

This URI can be used to update the errorCodeProfiles that is used in Overload Control feature for populating details in error responses when a request is discarded. By default, the errorCodeProfiles remains null.

- 1. From the left navigation menu, navigate to **NSSF** and click the **Common Configuration** option.
- Click the Ingress Gateway option to configure the Ingress Gateway APIs.
- Click the Error Code Profiles option under Ingress Gateway.The Error Code Profiles page is displayed.
- Click Add to add the profiles.
   The Add Error Code Profiles page is displayed.
- 5. Configure the following fields in the **Add Error Code Profiles** page:



- **a.** Name: Enter the name for the error profile. This name is also used while create Create Overload Control Discard Policies .
- b. Error Code: Enter the HttpStatusCode. This field in the errorScenario determines the HttpStatusCode that needs to be populated in ProblemDetails (HttpStatus field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType field.
- c. Error Cause: Enter the error cause details. This field in the errorScenario determines the error cause that needs to be populated in ProblemDetails (Cause field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter.
- d. Error Title: Enter the error title. This field in the errorScenario determines the title that needs to be populated in ProblemDetails (Title field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter.
- e. Redirect URL: Enter the redirection URL. This value is populated in LOCATION header while sending response from Ingress Gateway. The header is populated only when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter, the errorCode configured for the particular errorScenario lies in 3xx error series and the redirectUrl field for the particular errorScenario is configured appropriately.
- f. Retry After: Enter the value in seconds or particular date after which the service should be retried, this value is populated in Retry-After header while sending response from Ingress Gateway.
- g. Error Description: Enter the description that needs to be populated in ProblemDetails (Detail field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType field.
- Click Save on the Add Error Code Profiles page to save the details. Click Cancel to discard your progress and go back to Add Error Code Profiles page.

- Use Edit icon available in the next column of the specific entry to update the Error Code Profiles information.
- Use Refresh icon to refresh the configuration.

For more information on recommended parameter values, range, default values, and whether they are mandatory, optional, or conditional, see *Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide*.

#### 5.4.2.2 Create Overload Control Discard Policies

This URI can be used to update discard policies that will be used in overload control to select the appropriate policy from the configured list based on the load level of a particular service. By default, ocDicardPolicies is null.

- From the left navigation menu, navigate to NSSF and click the Common Configuration option.
- 2. Click the **Ingress Gateway** option to configure the Ingress Gateway APIs.



- 3. Click the Overload Control Discard Policies option under Ingress Gateway. The Overload Control Discard Policies page is displayed.
- Click Add to add the configuration.
   The Create Overload Control Discard Policies page is displayed.
- 5. Configure the following fields in the Create Overload Control Discard Policies page:
  - a. Name: Enter the name of the discarded policy.
  - b. **Scheme**: Enter the discarded policy scheme based on percentage.
  - c. Click Add in the Policies section.

The **Add Policies** page is displayed.

- d. Configure the following fields under the **Add Policies** page:
  - Value: Enter the value of priority above which requests are considered as potential candidates for drop. It is the percentage of requests to drop in the current sampling period over the calculated rate in the previous sampling period.
  - ii. **Action**: Enter the action to be taken on selected requests rejection based on error code. For example, RejectWithErrorCode.
  - iii. Level: Enter the overload level.
  - iv. **Error Code Profile**: Enter the name of the error code profile created in <u>Error Code Profiles</u>.
  - v. Click **Save** on the **Add Policies** page to save the details. Click **Cancel** to discard your progress and go back to **Add Policies** page.
- Click Save on the Create Overload Control Discard Policies page to save the details.
   Click Cancel to discard your progress and go back to Create Overload Control Discard Policies page.

#### (i) Note

 Use Edit or Delete icon available in the next column of the specific entry to update or delete the Policies information.

For more information on recommended parameter values, range, default values, and whether they are mandatory, optional, or conditional, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function REST Specification Guide*.

## 5.4.2.3 Discard Policy Mapping

This URI can be used to update service names and corresponding policy names for the service which is mapped to "ocDiscardPolicies" based on "policyName" and also to enable or disable the Overload Control feature and the sampling period in overload control. By default, the Overload Control feature is disabled and the sampling period is 6000.

- From the left navigation menu, navigate to NSSF and click the Common Configuration option.
- 2. Click the Ingress Gateway option to configure the Ingress Gateway APIs.
- 3. Click the **Discard Policy Mapping** option under **Ingress Gateway** to configure peers. The **Discard Policy Mapping** page is displayed with default configured values.



- Click Edit to update the Discard Policy Mapping.
   The Edit Discard Policy Mapping page is displayed.
- Configure the following fields in the Edit Discard Policy Mapping page:
  - Enabled: Use the switch to enable or disable discard policy mapping.
  - b. Mappings: Configure the following fields in Mappings section. Click Add to open Add Mappings window and add mapping details:
    - Service Name: Enter the service name. This field is used to determine a mapping between service and discard policy name per service name. It must be added in the following format:
      - <deployment-name>-<servicename>

**Note**: servicename is fixed and cannot be changed.

- Policy Name: Enter the policy name. It determines a mapping between the service and discards policy name per service.
- c. Click **Save** on the **Add Mappings** window to save the details. Click **Cancel** to discard your progress and go back to **Edit Discard Policy Mapping** page.

#### (i) Note

- Use Edit icon to edit an existing configuration.
- Use **Delete** icon to remove an existing configuration.
- **6. Sampling Period**: Add sampling period. It is the time frame for each cycle of Overload Control per service. Its value is in milliseconds.
- Click Save on the Edit Discard Policy Mapping page to save the details. Click Cancel to discard your progress and go back to Discard Policy Mapping page.

#### Note

Use Refresh icon to refresh the peer monitoring configuration.

For more information on recommended parameter values, range, default values, and whether they are mandatory, optional, or conditional, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function REST Specification Guide*.

#### 5.4.2.4 Error Code Series

This URI can be used to update the errorcodeseries list that are used in Overload Control feature and Server Header feature to list the configurable exception or error for an error scenario in Ingress Gateway.

- From the left navigation menu, navigate to NSSF and click the Common Configuration option.
- 2. Click the **Ingress Gateway** option to configure the Ingress Gateway APIs.
- Click the Error Code Series option under Ingress Gateway.The Error Code Seriespage is displayed.



- Click **Edit** to add the code series.
  - The **Edit Error Code Series** page is displayed.
- Configure the following fields in the **Edit Error Code Series**page:
  - a. ID: Enter an unique ID for error code.
  - b. Exception List: Lists the configurable exception or error for an error scenario in Ingress Gateway. The supported values are: ConnectionTimeout, RequestTimeout, UnknownHostException, ConnectException, RejectedExecutionException, InternalError, NotFoundException, ClosedChannelException, and BlackListIpException
- Click **Add** in the **Error Code Series** section. The **Add Error Code Series** page is displayed.
- Configure the following fields in the Add Error Code Series page:
  - Error Set: Enter the possible values include all error codes in the respective HttpSeries value assigned for "errorSet".
    - Note: Use single value of "-1" if all error codes in that HttpSeries are to be considered.
  - b. Error Codes: Enter the possible values for "errorSet" attribute: 5xx, 4xx, 3xx, 2xx, 1xx.
  - Click Save on the Add Error Code Series page to save the details. Click Cancel to discard your progress and go back to Add Error Code Series page.
- Click Save on the Edit Error Code Series page to save the details. Click Cancel to discard your progress and go back to **Edit Error Code Series** page.

- Use **Edit** icon available in the next column of the specific entry to update the **Error** Code Series information.
- Use **Refresh** icon to refresh the configuration.

For more information on recommended parameter values, range, default values, and whether they are mandatory, optional, or conditional, see Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.

## 5.4.2.5 Routes Configuration

The configuration of "routesconfiguration" is required for Server Header and Overload control feature to map route ID and its corresponding route-level configuration. By default, this configuration is null.

- From the left navigation menu, navigate to NSSF and click the Common Configuration option.
- 2. Click the **Ingress Gateway** option to configure the Ingress Gateway APIs.
- Click the **Routes Configuration** option under **Ingress Gateway**. The **Routes Configuration** page is displayed.
- 4. Click Add to add the configuration. The **Create Routes Configuration** page is displayed.
- Configure the following fields in the **Create Routes Configuration** page.



ID: Value of "id" attribute defines a specific service for route configuration. It specifies the route IDs for which you need to define server header.

#### Note

Ensure that route ID used in REST configuration is same as the route ID used in values.yaml file

- serverHeaderDetails: Configure the following fields in serverHeaderDetails section:
  - **Enabled**: Use the switch to enable or disable server header at route level.
  - **Error Code Series Id:** : Specify the error list ID.



#### Note

Ensure that an errorCodeSeries exists corresponding to the errorCodeSeriesId.

Click Save to save Routes configuration. Click Cancel to discard your progress, close the window, and go back to Create Routes Configuration page.

For more information on recommended parameter values, range, default values, and whether they are mandatory, optional, or conditional, see Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.

## 5.4.2.6 OAuth Validator Configurations

This REST API configuration is required for enabling access token validation using NRF Instance ID and key-ID (K-ID).

Before this configuration, perform the prerequisite steps and helm configuration explained in OAuth Access Token Based Authorization.

After Helm configuration, send the REST requests to use configured public key certificates. Using REST-based configuration, you can distinguish between the certificates configured on different NRFs and can use these certificates to validate the token received from a specific NRF.

- From the left navigation menu, navigate to NSSF and click the Common Configuration option.
- 2. Click the **Ingress Gateway** option to configure the Ingress Gateway APIs.
- Click the **OAuth Validator Configurations** option under **Ingress Gateway**. The **OAuth Validator Configurations** page is displayed.
- Click **Edit** to add the OAuth Validator Configurations. The **Edit OAuth Validator Configurations** page is displayed.
- Configure the following fields in the **Edit OAuth Validator Configurations** page:
  - Key ID List: Click Add in the Key ID List section and configure the following fields in Add Key ID List window:
    - Key ID: Enter the Key-ID.
    - Kubernetes Secret Name: Enter Kubernetes Secret Name.



- Kubernetes Secret Key For Certificate: Enter Kubernetes Secret Key for the certificate.
- Access Token Generation Algorithm: Enter the Access Token Generation Algorithm.
- b. Click Save on the Add Key ID List page to save the details. Click Cancel to discard your progress, close the window, and go back to Edit OAuth Validator Configurations page.
- c. Instance ID List: Click Add in the Instance ID List section and configure the following fields in Add Instance ID List window:
  - Instance ID: Enter the NRF Instance ID.
  - Kubernetes Secret Name: Enter the Kubernetes Secret Name.
  - Kubernetes Secret Key For Certificate: Enter Kubernetes Secret Key for the certificate.
  - Access Token Generation Algorithm: Enter the Access Token Generation Algorithm.
- Click Save on the Add Instance ID List page to save the details. Click Cancel to discard your progress, close the window, and go back to Edit OAuth Validator Configurations page.
- Access Token Validation Mode: Enter the mode of validation, which are INSTANCEID\_ONLY, KID\_ONLY, or KID\_PREFERRED. It will check for keyldList or instanceIdList for validation of token received based on mode selected.
- 8. Click **Save** on the **Edit OAuth Validator Configurations** page to save the details. Click **Cancel** to discard your progress and go back to **Edit OAuth Validator Configurations** page.

- Use Edit icon available in the next column of the specific entry to update the Key ID List or Instance ID List.
- Use Refresh icon to refresh the configuration.

For more information on recommended parameter values, range, default values, and whether they are mandatory, optional, or conditional, see *Oracle Communications Cloud Native Core*, *Network Slice Selection Function REST Specification Guide*.

#### 5.4.2.7 Server Header Details

This API can be used for adding Server Header in the error responses sent from Ingress Gateway. By default, this feature is disabled. To enable the feature, invoke the following REST API and update the enable switch.

- 1. From the left navigation menu, navigate to **NSSF** and click the **Common Configuration** option.
- 2. Click the **Ingress Gateway** option to configure the Ingress Gateway APIs.
- 3. Click the **Server Header Details** option under **Ingress Gateway** to configure peers. The **Server Header Details** page is displayed with default configured values.



- Click Edit to update the Server Header Details.
   The Edit Server Header Details page is displayed.
- 5. Configure the following fields in the **Edit Server Header Details** page:
  - **a. Enabled**: Use the switch to enable or disable Server Header.
  - **b. Error Code Series Id**: Specify the error list ID.

Ensure that an <u>errorCodeSeries</u> exists corresponding to the errorCodeSeriesId.

- c. Configuration: Configure the following fields in Configuration section:
  - NF Type: Specify the type of network function. In this case, it is NSSF.
  - NF Instance Id:: Enter the NSSF instance ID. It represents the UUID of the NSSF deployment that is used to generate the Server Header.
- Click Save on the Edit Server Header Details page to save the details. Click Cancel to discard your progress and go back to Server Header Details page.

#### 5.4.2.8 Pod Protection

This API is used to enable and configure Ingress Gateway Pod Protection Feature in NSSF. To enable the feature, REST API needs to be invoked and update the enabled flag to true.

Perform the following configurations:

- 1. From the left navigation menu, navigate to **NSSF** and click the **Common Configuration** option.
- Click the Ingress Gateway option to configure the Ingress Gateway APIs.
- Click the Pod Protection option under Ingress Gateway. The Pod Protection page is displayed.
- Click Edit to edit the configuration.
   The Edit Pod Protection page is displayed.
- 5. Configure the following fields in the **Edit Pod Protection** page:
  - **a. Enabled**: Use the switch to enable or disable Pod Protection feature. Setting this switch to enabled triggers resource monitoring in the next schedule.
  - b. Monitoring Interval: This attribute indicates the periodicity at which the overload state is monitored. Unit: Milliseconds



The proposed value for this attribute is 100. Minimum value is 0.

c. Enabled: Under Congestion Control Configurations, set the configuration of pod protection attributes for the Ingress Gateway pods.





This must be set to enabled if Pod Protection feature is enabled.

State Change sample count: This attribute indicates the number of times the pod must be in the particular congestion state before transitioning to another state. For example, if the current state is normal, and the new state is DoC, then NSSF moves the pod to DoC only if the state is reported for 10 times in 1 second (stateChangeSampleCount \* monitoringInterval).

#### (i) Note

The proposed value for this attribute is 10.

- Click **Add** in the **States** section to configure different States for Pod Protection. The **Add States** page is displayed.
- Configure the following fields in the **Add States** page:
  - **Name**: The name of the congestion state. Following are the proposed states:
    - Normal: The pod is not in overload state.
    - Danger of Congestion (Doc): The pod is about to go into the congested state. Actions configured in the entryAction is performed.
    - Congested state: The pod is in congested state. Actions configured in the entryAction is performed.
  - Weight: The weight of the congestion state. The weight indicates the critical of the congestion state. The lower the value, the lower the criticality.

#### (i) Note

The proposed value for this attribute is for Normal is 0, DoC is 1, and Congested is 2.

- iii. CPU: The CPU threshold is expressed in percentage. The proposed value for this attribute is for DoC is 75 and for Congested is 85.
- iv. Pending Message: The number of pending messages to be processed, expressed in absolute count. The proposed value for this attribute is for DoC is 1500 and for Congested is 2000.
- Click **Add** in the **Error Action** section. The **Add Error Action** page is displayed.
- vi. Configure the following fields in the **Add Error Action** page.
  - **Action**: This attribute indicates the action for the congestion state.
    - AcceptIncomingConnections: The action indicates whether the incoming new connection is accepted or rejected based on the overload state.
    - MaxConcurrentStreamsUpdate: The action indicates whether to increase or decrease the max concurrent stream for all incoming connections till the maxConcurrentStreamsPerCon is reached.
  - **Accept**: Use the switch to indicate if the incoming connection should be accepted or not. Applicable when the action is AcceptIncomingConnections.



- Enabled: The incoming connection is accepted.
- Disabled: The incoming connection is rejected.

The proposed value for Normal state is enabled, and for DoC and Congested state is disabled.

iii. **Increment by**: The attribute indicates the factor by which the current concurrent streams value will be incremented till it reaches maxConcurrentStreamsPerCon.

#### (i) Note

The proposed value for this attribute is 30.

iv. Increment by Action Sampling Period: The attribute indicates the time interval at which the incrementBy is applied to reach maxConcurrentStreamsPerCon. If not provided, the actionSamplingPeriod is used.

#### (i) Note

The proposed value for this attribute is 3. Unit is seconds.

v. **DecrementBy by**: The attribute indicates the factor by which the current concurrent streams value will be decremented till it reaches maxConcurrentStreamsPerCon.

#### Note

The proposed value for this attribute is 30.

vi. Decrement by Action Sampling Period: The attribute indicates the time interval at which the decrementBy is applied to reach maxConcurrentStreamsPerCon. If not provided, the actionSamplingPeriod is used.

#### (i) Note

The proposed value for this attribute is 1. Unit is seconds.

- vii. Max Concurrent Streams Per Con: The attribute indicates the maximum number of concurrent streams per connection allowed.

  The proposed value for this attribute is for Normal is 100, DoC is 10, and Congested is 1.
- viii. Click **Save** on the **Add Error Action** page to save the details. Click **Cancel** to discard your progress and go back to **Add Error Action** page.
- vii. Click **Save** on the **Add States**page to save the details. Click **Cancel** to discard your progress and go back to **Add States** page.
- g. Click **Save** on the **Edit Pod Protection**page to save the details. Click **Cancel** to discard your progress and go back to **Edit Pod Protection** page.



#### (i) Note

- Use **Edit** or **Delete** icon available in the next column of the specific entry to update or delete the States information.
- Use **Refresh** icon to refresh the configuration.

For more information on recommended parameter values, range, default values, and whether they are mandatory, optional, or conditional, see Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.

### 5.5 cnDBTier APIs

- 1. From the left navigation menu, navigate to **NSSF** and then click **cnDBTier** tab. The **cnDBTier** page is displayed.
- 2. Click **cnDBTier Health** to view the health status of the microservices like replication, backup manager, monitor services, and NDB services. The **cnDBTier Health** page is displayed.
  - Click the **Backup Manager Health Status** to view the health status of the backup manager.

The Backup Manager Health Status page is displayed.



#### (i) Note

The following APIs are read-only.

Table 5-21 Backup Manager Health Status

Fields	Description
Service Name	This attribute displays the service name of the backup manager microservice.
Service Status	This attribute displays the service status of the backup manager microservice. Possible values are UP and DOWN.
DB Connection Status	This attribute displays the database connection status of the backup manager microservice. Possible values are UP and DOWN.
Overall Backup Manager Service Health	This attribute displays the overall health status of the backup manager microservice. Possible values are UP and DOWN.
Backup Executor Health Status	This attribute displays the following information like node id and DB connection status of the backup executor.
Node Id	This attribute displays the id of the node.
DB Connection Status	This attribute displays the backup executor database connection status with the nodes. Possible values are UP and DOWN.

Click the Monitor Health Status to view the health status of the services.



The Monitor Health Status page is displayed.



#### (i) Note

The following APIs are read-only.

Table 5-22 Monitor Health Status details

Attribute	Description
Service Name	This attribute displays the service name of the monitor microservice.
DB Connection Status	This attribute displays the database connection status of the monitor microservice. Possible values are UP and DOWN.
Metric Scrape Status	This attribute displays the status of the metric scrape, that is if the metrics are fetched or not. If the metrics are fetched then the service is up and vice versa.  Possible values are UP and DOWN.
Overall Monitor Service Health	This attribute displays the overall health status of the monitor microservice. Possible values are UP and DOWN.

Click the NDB Health Status to view the health status of the network database. The **NDB Health Status** page is displayed.



#### (i) Note

The following APIs are read-only.

Table 5-23 NDB Health Status details

Attribute	Description
Local Site Name	This attribute displays the name of the current site. For example, site 1, site 2.
NDB Health Status Details	This attribute displays the health status of the network database like name of the NDB service, status of the service, health status of PVC.
Service Name	This attribute displays the service name. For example, ndbmgmd-0, ndbmtd-0, ndbmyappsqld-1, ndbmysqld-2.
Service Status	This attribute displays the status of the service. Possible values are UP and DOWN.
PVC Health Status	This attribute displays the health status of the PVC. Possible values are UP, DOWN, and NA.
	<b>Note</b> : This attribute is set to NA when some of the database pods are not connected to the PVC.



Click the **Replication Health Status** to view the health status of the replication sites. The **Replication Health Status** page is displayed.

#### (i) Note

The following APIs are read-only.

Table 5-24 Replication Health Status details

Attribute	Description
Local Site Name	This attribute displays the name of the current site (site 1, site 2).
Health Status Details	This attribute displays the health status details of the local site like replication service name, replication service status, database connection status of the replication service, and the overall health status of the replication micorservices. The number of rows in this table varies depending on the type of deployment (for example, two-site, three-site deployments).
Service Name	This attribute displays the name of the available replication service.
Service Status	This attribute displays the status of the available replication service. Possible values are UP and DOWN.
DB Connection Status	This attribute displays the database connection status of the replication microservice. Possible values are UP and DOWN.
Overall Replication Service Health	This attribute displays the overall health status of the replication microservice. Possible values are UP and DOWN.

Click **cnDBTier Version** to view the version. The **cnDBTier Version** page is displayed.



#### (i) Note

The following APIs are read-only.

Table 5-25 cnDBTier Version Attributes

Attribute	Description
cnDBTier Version	This attribute displays the cnDBTier version.
NDB Version	This attribute displays the network database (NDB) version.

Click the Backup List to view the list of completed backups along with backup ID, backup size, and backup creation timestamp.

The Backup List page is displayed.





#### (i) Note

The following APIs are read-only.

Table 5-26 Backup List

Fields	Description
Site Name	This attribute displays the name of the current site to which NSSF is connected.
Backup Details	This attribute displays the following information like backup id, backup size, and backup creation timestamp.
Backup Id	This attribute displays the ID of the stored backup.
Backup Size (bytes)	This attribute displays the size of the stored backup.
Creation TimeStamp	This attribute displays the time recorded when the backup was stored.

Click the **Database Statistics Report** to view the available databases. The **Database Statistics Report** page is displayed.



#### (i) Note

The following APIs are read-only.

**Table 5-27 Database Statistics Report** 

Fields	Description
Database Count	This attribute displays the number of available database.
Database Tables Count	This attribute displays the available database names and their table count.
Database Name	This attribute displays the database name.
Table Count	This attribute displays the table count for each database.
Database Table Rows Count	This attribute displays the table rows present in each table.
Database Name	This attribute displays the database name.

Click on the View icon available next to the database name to view the View Database Table Rows Count screen.

The View Database Table Rows Count page is displayed.

Table 5-28 View Database Table Rows Count

Fields	Description
Database Name	This attribute displays the database name.
Tables	This attribute displays the table names and the corresponding rows in each table.



Table 5-28 (Cont.) View Database Table Rows Count

Fields	Description
Table Name	This attribute displays the table name.
Row Count	This attribute displays the table rows present in each table.

6. Click Georeplication Status to view the local site and remote site name to which NSSF is connected.

The Georeplication Status page is displayed.



#### Note

The following APIs are read-only.

**Table 5-29 GeoReplication Status** 

Attribute	Description
Local Site Name	This attribute displays the local site name to which NSSF is connected.
	<b>Note:</b> The number of local site names may vary depending on the type of georeplication used in NSSF.
Remote Site Name	This attribute displays the remote site name.
	<b>Note:</b> The number of remote site names may vary depending on the type of georeplication used in NSSF.
Replication Status	This attribute displays the replication status with corresponding sites.
	<b>Note:</b> The number of replication statuses may vary depending on the type of georeplication used in NSSF.
Seconds Behind Remote Site	This attribute displays the number of seconds that the last record read by the local site is behind the latest record written by the remote site for all the replication groups.  Note: The number of replication statuses may vary depending on the type of georeplication used in NSSF.

Click on the View icon in the Actions menu, to view the View Georeplication Status screen.

The **Georeplication Status** page is displayed.

Table 5-30 Georeplication Status

Attribute	Description
Replication Group Delay	This attribute displays the seconds behind the remote site for individual replication groups.
Replication Channel Group Id	This attribute displays the ID of the replication channel group.

b. Click on the View icon to view the Replication Group Delay attributes. The Replication Group Delay page is displayed.



**Table 5-31 View Replication Group Delay** 

Attribute	Description
Channel Details	This attribute displays the channel details such as Remote Replication IP and Role.
Remote Replication IP	This attribute displays the IP of the remote replication channel.
Role	This attribute displays the role of the replication channel IP.

7. Click the HeartBeat Status to view the connectivity between local site and remote site name to which NSSF is connected.

The **HeartBeat Status** page is displayed.



#### Note

The following APIs are read-only.

Table 5-32 HeartBeat Status Details

Fields	Description
Site Name	This attribute displays the name of the current site to which NSSF is connected.
HeartBeat Details	This attribute displays the following information like remote site name, heart beat status, heart beat lag, and replication channel group id.
Remote Site Name	This attribute displays the remote site name.
Heartbeat Status	This attribute displays the connectivity status with corresponding sites.
Heartbeat Lag	This attribute displays the lag or latency in seconds it took to synchronize between sites.
Replication channel Group Id	This attribute displays the ID of the replication channel group.

Click **Local Cluster Status** to view the local cluster status for the current site. The Local Cluster Status page is displayed.



#### (i) Note

The following APIs are read-only.

**Table 5-33 Local Cluster Status** 

Attribute	Description
Site Name	This attribute displays the name of the current site to which NSSF is connected.
Cluster Status	This attribute displays the local cluster status for the current site.

Click the On Demand Backup to create a new backup and view the status of initiated ondemand backups.



#### The **On Demand Backup** page is displayed.

#### (i) Note

The following APIs are read-only.

Table 5-34 On Demand Backup Details

Fields	Description
Site Name	This attribute displays the name of the current site to which NSSF is connected.
DR Status	This attribute displays the disaster recovery status.
Backup Id	This attribute displays the ID of the stored backup.
Backup Status	This attribute displays the status of backup.
Remote Transfer Status	The attribute displays the status of remote transfer.
Initiate Backup	The attribute displays whether the backup is initiated or not.  Note: You can read and write this API.

Click Edit. The Edit On Demand Backup page appears.



#### (i) Note

The **Edit** mode is available only for Initiate Backup.

- Use the Toggle option to Initiate the backup and click Save. A confirmation message "Save successfully" appears.
- Click Cancel to navigate back to the On Demand Backup page.
- Click **Refresh** to reload the On Demand Backup page.

# NSSF Metrics, KPIs, and Alerts

This chapter includes information about Metrics, KPIs, and Alerts for Oracle Communications Cloud Native Core, Network Slice Selection Function.



#### (i) Note

The performance and capacity of the NSSF system may vary based on the call model, Feature or Interface configuration, and underlying CNE and hardware environment.

## 6.1 NSSF Metrics

This section includes information about dimensions, common attributes, and metrics for NSSF.

#### **Metric Types**

The following table describes the NSSF metric types used to measure the health and performance of NSSF and its core functionalities:

Table 6-1 Metric Type

Metric Type	Description
Counter	Represents the total number of occurrences of an event or traffic, such as measuring the total amount of traffic received and transmitted by SCP, and so on.
Gauge	Represents a single numerical value that changes randomly. This metric type is used to measure various parameters, such as SCP load values, memory usage, and so on.
Histogram	Represents samples of observations (such as request durations or response sizes) and counts them in configurable buckets. It also provides a sum of all observed values.

#### **Dimensions**

The following table describes different types of metric dimensions:

Table 6-2 Dimensions

Dimension	Description	Values
retryCount	The attempt number to send a notification.	Depends on the helm parameter httpMaxRetries (1, 2)
ResponseCode	HTTP response code.	Bad Request, Internal Server Error etc. (HttpStatus.*)
CauseCode	It specifies the cause code of an error response.	Cause Code of the error response. For example, "SUBSCRIPTION_NOT_FOUND"
Message Type	This specifies the type of NS-Selection query message.	INITIAL_REGISTRATION/PDU_SESSION/ UE_CONFIG_UPDATE



Table 6-2 (Cont.) Dimensions

Dimension	Description	Values
	Description	
NFType	It specifies the name of the NF Type.	For example: Path is /nxxx-yyy/vz/ Where XXX(Upper Case) is NFType UNKNOWN if unable to extract NFType from the path.
NFServiceType	Name of the Service within the NF.	For Eg: Path is /nxxx-yyy/vz/ Where nxxx-yyy is NFServiceType UNKNOWN if unable to extract NFServiceType from the path.
Host	Specifies IP or FQDN port of ingress gateway.	NA
HttpVersion	Specifies Http protocol version.	HTTP/1.1, HTTP/2.0
Scheme	Specifies the Http protocol scheme.	HTTP, HTTPS, UNKNOWN
ClientCertIdentity	Cerificate Identity of the client.	SAN=127.0.0.1,localhost CN=localhost, N/A if data is not available
Route_Path	Path predicate or Header predicate that matches the current request.	NA
Instanceldentifier	Prefix of the pod configured in helm when there are multiple instances in same deployment.	Prefix configured in helm, UNKNOWN
ErrorOriginator	Captures the ErrorOriginator.	ServiceProducer, Nrf, IngresGW, None
quantile	Captures the latency values with ranges as 10ms, 20ms, 40ms, 80ms, 100ms, 200ms, 500ms, 1000ms and 5000ms.	Integer values
releaseVersion	Indicates the current release version of Ingress or Egress gateway.	Picked from helm chart {{ .Chart.Version }}
configVersion	Indicates the configuration version that Ingress or gateway is currently maintaining.	Value received from config server (1, 2)
updated	Indicates whether the configuration is updated or not.	True, False
Direction	Indicates the direction of connection established, that is, whether it is incoming or outgoing.	ingress, egressOut
AMF Instance Id	NF-Id of AMF	NA
Subscription- Id	Subscription -ID	NA
Operation	NSAvailability Operation	UPDATE/DELETE/SUBSCRIBE/UNSUBSCRIBE
Method	HTTP method	POST/PUT/PATCH/DELETE/GET/OPTIONS
Status	HTTP response code	NA
query_type	Type of DB read query	applypolicy_reg/applypolicy_pdu/evaluate_amfset/ evaluate_resolution
ConsumerNFInstanceId	NF instance id of the NF service consumer.	NA
ConsumerNFType	The NF type of the NF service consumer.	NRF, UDM, AMF, SMF, AUSF, NEF, PCF, SMSF, NSSF, UDR, LMF, GMLC,5G_EIR, SEPP, UPF, N3IWF, AF, UDSF, BSF, CHF, NWDAF
TargetNFType	The NF type of the NF service producer.	NRF, UDM, AMF, SMF, AUSF, NEF, PCF, SMSF, NSSF, UDR, LMF, GMLC,5G_EIR, SEPP, UPF, N3IWF, AF, UDSF, BSF, CHF, NWDAF



### Table 6-2 (Cont.) Dimensions

Dimension	Description	Values
TargetNFInstanceId	NF instance ID of the NF service producer	NA
scope	NF service name(s) of the NF service producer(s), separated by whitespaces.	NA
StatusCode	Status code of NRF access token request.	Bad Request, Internal Server Error etc. (HttpStatus.*)
issuer	NF instance ID of NRF	NA
subject	NF instance ID of service consumer	NA
reason	The reason contains the human readable message for oauth validation failure.	NA
ConfigurationType	Determines the type of configuration in place for OAuth Client in Egress Gateway. If nrfClientQueryEnabled Helm parameter in oauthClient Helm configurations at Egress Gateway is false then the ConfigurationType is STATIC, else DYNAMIC.	STATIC, DYNAMIC
id	Determines the keyid or instance id that is configured via persistent configuration when oauth is enabled.	NA
certificateName	Determines the certificate name inside a specific secret that is configured via persistent configuration when oauth is enabled.	NA
secretName	Determines the secret name that is configured via persistent configuration when oauth is enabled	NA
Source	Determines if the configuration is done by the operator or fetched from AMF.	OperatorConfig/LearnedConfigAMF
ERRORTYPE	Determines the type of error.	DB_ERROR/MISSING_CONFIGURATION/ UNKNOWN
NegotiatedTLSVersion	This denotes the TLS version used for communication between the server and the client.	TLSv1.2, TLSv1.3.
serialNumber	Indicates the type of the certificate.	serialNumber=4661 is used for RSA and serialNumber =4662 is used for ECDSA
NrfUri	URI of the Network Repository Function Instance.	For example: nrf-stubserver.ocnssf-site:8080



Table 6-2 (Cont.) Dimensions

Dimension	Description	Values
Subscription_removed	The dimension indicates the status of a subscription upon receiving a 404 response from the AMF after a	"false": The subscription was not deleted. This value applies if the feature is disabled, indicating no deletion attempt was made.
	notification is sent.	"true": The subscription was successfully deleted. This value applies if the feature is enabled and the deletion process completed successfully.
		"error": The subscription was not deleted due to internal issues, such as a database error, despite the feature being enabled and a deletion attempt being made.
VirtualFqdn	FQDN that shall be used by the alternate service for the DNS lookup	Valid FQDN

#### **Common Attributes**

The following table includes information about common attributes for NSSF.

Table 6-3 Common Attributes

Attribute	Description
application	The name of the application that the microservice is a part of.
eng_version	The engineering version of the application.
microservice	The name of the microservice.
namespace	The namespace in which microservice is running.
node	The name of the worker node that the microservice is running on.

## 6.1.1 NSSF Success Metrics

This section provides details about the NSSF success metrics.

Table 6-4 ocnssf\_nsselection\_rx\_total

Field	Details
Description	Count of request messages received by NSSF for the Nnssf_NSSelection service.
Туре	Counter
Service Operation	NSSelection
Dimension	<ul><li>AMF Instance Id</li><li>Message Type</li><li>plmn</li></ul>

Table 6-5 ocnssf\_nsselection\_success\_tx\_total

Field	Details
Description	Count of success response messages sent by NSSF for requests for the Nnssf_NSSelection service.



#### Table 6-5 (Cont.) ocnssf\_nsselection\_success\_tx\_total

Field	Details
Туре	Counter
Service Operation	NSSelection
Dimension	<ul><li>AMF Instance Id</li><li>Message Type</li><li>plmn</li></ul>

#### Table 6-6 ocnssf\_nsselection\_policy\_match\_total

Field	Details
Description	Count of policy matches found during processing of request messages for the Nnssf_NSSelection service.
Туре	Counter
Service Operation	NSSelection
Dimension	<ul> <li>AMF Instance Id</li> <li>Message Type</li> <li>Policy Rule Profile</li> <li>plmn</li> </ul>

#### Table 6-7 ocnssf\_nsselection\_time\_match\_total

Field	Details
Description	Count of time profile matches found during processing of request messages for the Nnssf_NSSelection service.
Туре	Counter
Service Operation	NSSelection
Dimension	<ul> <li>AMF Instance Id</li> <li>Message Type</li> <li>Time Profile Name</li> <li>plmn</li> </ul>

#### Table 6-8 ocnssf\_nsselection\_nsi\_selected\_total

Field	Details
Description	Count of NRF discoveries performed during processing of request messages for the Nnssf_NSSelection service.
Туре	Counter
Service Operation	NSSelection
Dimension	<ul> <li>AMF Instance Id</li> <li>Message Type</li> <li>NSI Profile Name</li> <li>plmn</li> </ul>



#### Table 6-9 ocnssf\_nsavailability\_notification\_trigger\_tx

Field	Details
Description	Count of notification triggers sent to NsSubscription.
Туре	Counter
Service Operation	NSAvailability
Dimension	Method

#### Table 6-10 ocnssf\_nsavailability\_notification\_trigger\_response\_rx

Field	Details
Description	Count of success response for notification trigger by NSSubscription.
Туре	Counter
Service Operation	NSAvailability
Dimension	Method

#### Table 6-11 ocnssf\_nsselection\_nrf\_disc

Field	Details
Description	Count of NRF discoveries performed during processing of request messages for the Nnssf_NSSelection service.
Туре	Counter
Service Operation	NSSelection
Dimension	None

#### Table 6-12 ocnssf\_nsselection\_nrf\_disc\_success

Field	Details
Description	Count of successful discovery results received from NRF during processing of request messages for the Nnssf_NSSelection service.
Туре	Counter
Service Operation	NSSelection
Dimension	None

#### Table 6-13 ocnssf\_nssaiavailability\_rx\_total

Field	Details
Description	Count of request messages received by NSSF for the Nnssf_NSSAlAvailability service.
Туре	Counter
Service Operation	NSAvailability
Dimension	• Nfld
	Method
	Message Type



#### Table 6-14 ocnssf\_nssaiavailability\_success\_tx\_total

Field	Details
Description	Count of success response messages sent by NSSF for requests for the Nnssf_NSSAlAvailability service.
Туре	Counter
Service Operation	NSAvailability
Dimension	<ul><li>Nfld</li><li>Method</li><li>Message Type</li></ul>

#### Table 6-15 ocnssf\_nssaiavailability\_options\_rx

Details
Count of HTTP options received at NSAvailability service.
Counter
NSAvailability
<ul><li>Method</li><li>Message Type</li></ul>

#### Table 6-16 ocnssf\_nssaiavailability\_options\_tx\_status\_ok

Field	Details
Description	Count of HTTP options response with status 200 OK.
Туре	Counter
Service Operation	NSAvailability
Dimension	Method
	Message Type

#### Table 6-17 ocnssf\_nssaiavailability\_notification\_indirect\_communication\_rx\_total

Field	Details
Description	Count of request notification messages sent by NSSF using indirect communication.
Туре	Counter
Service Operation	NSAvailability
Dimension	Method
	Message Type

#### Table 6-18 ocnssf\_nssaiavailability\_notification\_indirect\_communication\_tx\_total

Field	Details
Description	Count of notification response messages received by NSSF using indirect communication.
Туре	Counter
Service Operation	NSAvailability
Dimension	Method
	Message Type



#### Table 6-19 ocnssf\_nssaiavailability\_indirect\_communication\_rx\_total

Field	Details
Description	Count of request when subscription messages received by NSSF using indirect communication.
Туре	Counter
Service Operation	NSAvailability
Dimension	<ul><li>Message Type</li><li>Method</li></ul>

#### Table 6-20 ocnssf\_nssaiavailability\_indirect\_communication\_tx\_total

Field	Details
Description	Count of subscription response messages sent by NSSF using indirect communication.
Туре	Counter
Service Operation	NSAvailability
Dimension	<ul><li>Message Type</li><li>Method</li></ul>

#### Table 6-21 ocnssf\_nsselection\_requests\_duration\_seconds\_sum

Field	Details
Description	Time duration in seconds taken by NSSF to process requests to NSSelection.
Туре	Counter
Service Operation	NSSelection
Dimension	None

#### Table 6-22 ocnssf\_nsselection\_requests\_duration\_seconds\_count

Field	Details
Description	Count of number of requests processed by NSSelection.
Туре	Counter
Service Operation	NSSelection
Dimension	None

#### Table 6-23 ocnssf\_nsselection\_requests\_duration\_seconds\_max

Field	Details
Description	Maximum time duration in seconds taken by NSSF to process requests to NSSelection.
Туре	Counter
Service Operation	NSSelection
Dimension	None



#### Table 6-24 ocnssf\_db\_query\_duration\_seconds\_sum

Field	Details
Description	Time duration in seconds to process dbQuery.
Туре	Counter
Service Operation	NA
Dimension	query_type

#### Table 6-25 ocnssf\_db\_query\_duration\_seconds\_count

Field	Details
Description	Count of number of dbQuery.
Туре	Counter
Service Operation	NA
Dimension	query_type

#### Table 6-26 ocnssf\_db\_query\_duration\_seconds\_max

Field	Details
Description	Maximum time duration in seconds taken to process dbQuery.
Туре	Counter
Service Operation	NA
Dimension	query_type

#### Table 6-27 ocnssf\_nssaiavailability\_submod\_rx\_total

Field	Details
Description	Count of HTTP patch for subscription (SUBMOD) request messages received by NSSF for ocnssf_NSSAIAvailability service.
Туре	Counter
Service Operation	NSAvailability
Dimension	<ul><li>SubscriptionId</li><li>Method</li></ul>

#### Table 6-28 ocnssf\_nssaiavailability\_submod\_success\_response\_tx\_total

Field	Details
Description	Count of success response messages sent by NSSF for HTTP patch for subscription (SUBMOD) requests for ocnssf_NSSAlAvailability service.
Туре	Counter
Service Operation	NSAvailability
Dimension	<ul><li>SubscriptionId</li><li>Method</li><li>Return code</li></ul>



#### Table 6-29 ocnssf\_nssaiavailability\_notification\_success\_response\_rx\_total

Field	Details
Description	Count of success notification response messages received by NSSF for requests for the Nnssf_NSSAIAvailability service.
Туре	Counter
Service Operation	NSSubscription
Dimension	<ul><li>Method</li><li>Message Type</li></ul>

#### Table 6-30 ocnssf\_nssaiavailability\_notification\_tx\_total

Field	Details
Description	Count of notification messages sent by NSSF as part of Nnssf_NSSAIAvailability service.
Туре	Counter
Service Operation	NSSubscription
Dimension	Method
	Message Type

#### Table 6-31 ocnssf\_notification\_trigger\_rx\_total

Field	Details
Description	Count of notification triggers received by NSSF.
Туре	Counter
Service Operation	NSSubscription
Dimension	<ul><li>Trigger type</li><li>Method</li></ul>

#### Table 6-32 ocnssf\_nsconfig\_notification\_trigger\_tx\_total

Field	Details
Description	Count of notification triggers sent to NsSubscription.
Туре	Counter
Service Operation	NSConfig
Dimension	Message Type
	Method

#### Table 6-33 ocnssf\_nsconfig\_notification\_trigger\_response\_rx\_total

Field	Details
Description	Count of success response for notification trigger by NsSubscription.
Туре	Counter
Service Operation	NSConfig
Dimension	Method



Table 6-34 ocnssf\_nsconfig\_nrf\_disc\_success\_total

Field	Details
Description	Count of successful discovery results received from NRF during processing of configuration of amf_set in Nnssf_NSConfig service.
Туре	Counter
Service Operation	NSConfig
Dimension	None

Table 6-35 ocnssf\_subscription\_nrf\_tx\_total

Field	Details
Description	Count of successful subscription results received from NRF during processing of configuration of amf_set in Nnssf_NSConfig service.
Туре	Counter
Service Operation	NSConfig
Dimension	None

## 6.1.2 NSSF Error Metrics

This section provides details about the NSSF error metrics.

Table 6-36 ocnssf\_configuration\_database\_read\_error

Field	Details
Description	Count of errors encountered when trying to read the configuration database.
Туре	Counter
Service Operation	NSSelection
Dimension	None

Table 6-37 ocnssf\_configuration\_database\_write\_error

Field	Details	
Description	Count of errors encountered when trying to write to the configuration database.	
Туре	Counter	
Service Operation	NSConfig	
Dimension	None	

Table 6-38 ocnssf\_nsconfig\_notification\_trigger\_failure\_response\_rx\_total

Field	Details
Description	Count of failure response for notification trigger by NSSubscription.
Туре	Counter
Service Operation	NSConfig
Dimension	<ul><li>Message Type</li><li>Method</li></ul>



#### Table 6-39 ocnssf\_nsconfig\_notification\_trigger\_retry\_tx\_total

Field	Details
Description	Count of retry notification triggers sent to NSSubscription.
Туре	Counter
Service Operation	NSConfig
Dimension	<ul><li>Message Type</li><li>Method</li></ul>

#### Table 6-40 ocnssf\_nsconfig\_notification\_trigger\_failed\_tx\_total

Field	Details
Description	Count of failed notification triggers (all retrys failed) to NSSubscription.
Туре	Counter
Service Operation	NSConfig
Dimension	<ul><li>Message Type</li><li>Method</li></ul>

#### Table 6-41 ocnssf\_nsconfig\_nrf\_disc\_error\_total

Field	Details
Description	Count of failed discovery results received from NRF during processing of configuration of amf_set in Nnssf_NSConfig service.
Туре	Counter
Service Operation	NSConfig
Dimension	None

#### Table 6-42 ocnssf\_discovery\_nrf\_tx\_failed\_total

Field	Details
Description	Count of failed discovery requests sent by NSSF to NRF during configuration of amf_set in Nnssf_NSConfig service.
Туре	Counter
Service Operation	NSConfig
Dimension	None

#### Table 6-43 ocnssf\_subscription\_nrf\_tx\_failed\_total

Field	Details
Description	Count of failed subscription results received from NRF during processing of configuration of amf_set in Nnssf_NSConfig service.
Туре	Counter
Service Operation	NSConfig
Dimension	None



#### Table 6-44 ocnssf\_state\_data\_read\_error

Field	Details
Description	Count of errors encountered when trying to read the state database.
Туре	Counter
Service Operation	NSSelection
Dimension	None

#### Table 6-45 ocnssf\_state\_data\_write\_error

Field	Details
Description	Count of errors encountered when trying to write to the state database.
Туре	Counter
Service Operation	NSAvailability
Dimension	None

#### Table 6-46 ocnssf\_nsselection\_nrf\_disc\_failure\_total

Field	Details
Description	Count of errors encountered when trying to reach the NRF's discovery service.
Туре	Counter
Service Operation	NSSelection
Dimension	None

#### Table 6-47 ocnssf\_nsselection\_policy\_not\_found\_total

Field	Details
Description	Count of request messages that did not find a configured policy.
Туре	Counter
Service Operation	NSSelection
Dimension	<ul><li>AMF Instance Id</li><li>Message Type</li><li>plmn</li></ul>

#### Table 6-48 ocnssf\_nsselection\_unsupported\_plmn\_total

Field	Details
Description	Count of request messages that did not find mcc and mnc in the PLMN list.
Туре	Counter
Service Operation	NSSelection
Dimension	None



#### Table 6-49 ocnssf\_nssaiavailability\_subscription\_failure\_total

Field	Details
Description	Count of subscribe requests rejected by NSSF.
Туре	Counter
Service Operation	NSAvailability
Dimension	None

#### Table 6-50 ocnssf\_nssaiavailability\_notification\_error\_response\_rx\_total

Field	Details
Description	Count of failure notification response messages received by NSSF for requests by the Nnssf_NSSAIAvailability service.
Туре	Counter
Service Operation	NSSubscription
Dimension	<ul> <li>MessageType</li> <li>Method</li> <li>ResponseCode</li> <li>CauseCode</li> <li>retryCount</li> </ul>

#### Table 6-51 ocnssf\_nssaiavailability\_notification\_failure

Field	Details
Description	Count of failure notification response messages received by NSSF for requests by the Nnssf_NSSAI Availability service.
Туре	Counter
Service Operation	NSSubscription
Dimension	<ul><li>Subscription- Id</li><li>Status</li></ul>

#### Table 6-52 ocnssf\_nssaiavailability\_options\_tx\_status\_unsupportedmediatype

Field	Details
Description	Count of HTTP OPTIONS response with status 415 Unsupported Media type.
Туре	Counter
Service Operation	NSAvailability
Dimension	<ul><li>Message Type</li><li>Method</li></ul>

#### Table 6-53 ocnssf\_nsavailability\_unsupported\_plmn\_total

Field	Details
Description	Count of request messages with unsupported PLMN received by NSSF for the ocnssf_NSAvailability service.
Туре	Counter
Service Operation	NSAvailability



#### Table 6-53 (Cont.) ocnssf\_nsavailability\_unsupported\_plmn\_total

Field	Details
Dimension	Message Type
	Method

#### Table 6-54 ocnssf\_nsavailability\_invalid\_location\_url\_total

Field	Details
Description	Count of invalid location header.
Туре	Counter
Service Operation	NSAvailability
Dimension	<ul><li>Message Type</li><li>Method</li></ul>

#### Table 6-55 ocnssf\_nssaiavailability\_submod\_error\_response\_tx\_total

Field	Details
Description	Count of error response messages sent by NSSF for HTTP patch for subscription (SUBMOD) requests for ocnssf_NSSAIAvailability service.
Туре	Counter
Service Operation	NSAvailability
Dimension	<ul><li>ReturnCode</li><li>SubscriptionId</li><li>Method</li></ul>

#### Table 6-56 ocnssf\_nssaiavailability\_submod\_unimplemented\_op\_total

Field	Details
Description	Count of HTTP patch request messages received by NSSF for ocnssf_NSSAIAvailability service for which PATCH operation (op) is not implemented.
Туре	Counter
Service Operation	NSAvailability
Dimension	<ul><li>ReturnCode</li><li>SubscriptionId</li><li>Method</li></ul>

#### Table 6-57 ocnssf\_nssaiavailability\_submod\_patch\_apply\_error\_total

Field	Details
Description	Count of HTTP patch request messages received by OCNSSFfor ocnssf_NSSAlAvailability service for which PATCH application returned error.
Туре	Counter
Service Operation	NSAvailability
Dimension	<ul><li>ReturnCode</li><li>SubscriptionId</li><li>Method</li></ul>



#### Table 6-58 ocnssf\_nsavailability\_notification\_trigger\_failure\_response\_rx

Field	Details	
Description	Count of failure response for notification trigger by NSSubscription.	
Туре	Counter	
Service Operation	NSAvailability	
Dimension	Method	

#### Table 6-59 ocnssf\_nsavailability\_notification\_trigger\_retry\_tx

Field	Details
Description	Count of retry notification triggers sent to NSSubscription.
Туре	Counter
Service Operation	NSAvailability
Dimension	Method

#### Table 6-60 ocnssf\_nsavailability\_notification\_trigger\_failed\_tx

Field	Details
Description	Count of failed notification triggers (all retries failed) to NSSubscription.
Туре	Counter
Service Operation	NSAvailability
Dimension	Method

#### Table 6-61 ocnssf\_nssaiavailability\_notification\_delete\_on\_subscription\_not\_found\_total

Field	Details
Description	Triggered when 404 Subscription with SUBSCRIPTION_NOT_FOUND is received by AMF.
Туре	Counter
Service Operation	NsSubscription
Dimension	Subscription_Removed

#### Table 6-62 ocnssf\_nssaiavailability\_notification\_db\_error

Field	Details
Description	Triggered when DB error or exception occurs when trying to delete NssaiSubscription.
Туре	Counter
Service Operation	NsSubscription
Dimension	None

#### Table 6-63 ocnssf\_nssaiavailability\_indirect\_communication\_subscription\_failure\_total

Field	Details
Description	Count of failure when subscription messages sent by NSSF using indirect communication.
Туре	Counter



Table 6-63 (Cont.) ocnssf\_nssaiavailability\_indirect\_communication\_subscription\_failure\_total

Field	Details
Service Operation	NSAvailability
Dimension	<ul><li>Message Type</li><li>Method</li></ul>

Table 6-64 ocnssf\_nssaiavailability\_indirect\_communication\_notification\_failure\_total

Field	Details
Description	Count of failure when notification messages sent by NSSF using indirect communication.
Туре	Counter
Service Operation	NSSubscription
Dimension	<ul><li>ReturnCode</li><li>Message Type</li><li>Method</li></ul>

## 6.1.3 NSSF Common metrics

This section provides details about the NSSF common metrics.

Table 6-65 security\_cert\_x509\_expiration\_seconds

Field	Details
Description	Indicates the time to certificate expiry in epoch seconds.
Туре	Histogram
Dimension	serialNumber

Table 6-66 http\_requests\_total

Field	Details
Description	This is pegged as soon as the request reaches the Ingress or Egress gateway in the first custom filter of the application.
Туре	Counter
Dimension	<ul> <li>direction: ingress or egress</li> <li>method: the method from the request line</li> <li>uri: the URI from the request line</li> <li>http_version: the HTTP version from the request line</li> <li>host: the value of the Host header field</li> <li>NFType</li> <li>NFServiceType</li> <li>HttpVersion</li> <li>Scheme</li> <li>Route_path</li> <li>InstanceIdentifier</li> <li>ClientCertIdentity</li> </ul>



#### Table 6-67 http\_responses\_total

Field	Details
Description	Responses received or sent from the microservice .
Туре	Counter
Dimension	<ul> <li>Status</li> <li>Method</li> <li>Route_path</li> <li>NFType</li> <li>NFServiceType</li> <li>Host</li> <li>HttpVersion</li> <li>Scheme</li> <li>InstanceIdentifier</li> <li>ClientCertIdentity</li> </ul>

### Table 6-68 http\_request\_bytes

Field	Details
Description	Size of requests, including header and body. Grouped in 100 byte buckets.
Туре	Histogram
Dimension	<ul><li>direction</li><li>method</li><li>uri</li><li>http_version</li></ul>

#### Table 6-69 http\_response\_bytes

Field	Details
Description	Size of responses, including header and body. Grouped in 100 byte buckets.
Туре	Histogram
Dimension	<ul><li>direction</li><li>http_version</li></ul>

#### Table 6-70 bandwidth\_bytes

Field	Details
Description	Amount of ingress and egress traffic sent and received by the microservice.
Туре	Counter
Dimension	direction

#### Table 6-71 request\_latency\_seconds

Field	Details
Description	This metric is pegged in the last custom filter of the Ingress or Egress gateway while the response is being sent back to the consumer NF. It tracks the amount of time taken for processing the request. It starts as soon the request reaches the first custom filter of the application and lasts till the response is sent back to the consumer NF from the last custom filter of the application.



#### Table 6-71 (Cont.) request\_latency\_seconds

Field	Details
Туре	Histogram
Dimension	<ul> <li>quantile</li> <li>InstanceIdentifier</li> <li>Route_path</li> <li>Method</li> </ul>

#### Table 6-72 connection\_failure\_total

Field	Details
Description	This metric is pegged by jetty client when the destination is not reachable by Ingress or Egress gateway. In case of Ingress gateway, the destination service will be a back-end microservice of the NF, and TLS connection failure metrics when connecting to ingress with direction as ingress. For Egress gateway, the destination is producer NF.
Туре	Counter
Dimension	<ul> <li>Host</li> <li>Port</li> <li>InstanceIdentifier</li> <li>Direction</li> <li>error_reason</li> </ul>

#### Table 6-73 request\_processing\_latency\_seconds

Field	Details
Description	This metric is pegged in the last custom filter of the Ingress or Egress gateway while the response is being sent back to the consumer NF. This metric captures the amount of time taken for processing of the request only within Ingress or Egress gateway. It starts as soon the request reaches the first custom filter of the application and lasts till the request is forwarded to the destination.
Туре	Timer
Dimension	<ul><li>quantile</li><li>InstanceIdentifier</li><li>Route_path</li><li>Method</li></ul>

#### Table 6-74 jetty\_request\_stat\_metrics\_total

Field	Details
Description	This metric is pegged for every event occurred when a request is sent to Ingress or Egress gateway.
Туре	Counter
Dimension	<ul><li>event</li><li>client_type</li><li>InstanceIdentifier</li></ul>



#### Table 6-75 jetty\_response\_stat\_metrics\_total

Field	Details
Description	This metric is pegged for every event occurred when a response is received by Ingress or Egress gateway.
Туре	Counter
Dimension	<ul><li>event</li><li>client_type</li><li>InstanceIdentifier</li></ul>

#### Table 6-76 server\_latency\_seconds

Field	Details
Description	This metric is pegged in Jetty response listener that captures the amount of time taken for processing of the request by jetty client
Туре	Timer
Dimension	<ul><li>quantile</li><li>InstanceIdentifier</li><li>Method</li></ul>

#### Table 6-77 roundtrip\_latency\_seconds

Field	Details
Description	This metric is pegged in Netty outbound handler that captures the amount of time taken for processing of the request by netty server.
Туре	Timer
Dimension	<ul><li>quantile</li><li>InstanceIdentifier</li><li>Method</li></ul>

#### Table 6-78 oc\_configclient\_request\_total

Field	Details
Description	This metric is pegged whenever config client is polling for configuration update from common configuration server.
Туре	Counter
Dimension	<ul><li>Release version</li><li>Config version</li></ul>

#### Table 6-79 oc\_configclient\_response\_total

Field	Details
Description	This metrics is pegged whenever config client receives response from common configuration server.
Туре	Counter
Dimension	<ul><li>Release version</li><li>Config version</li><li>Updated</li></ul>



#### Table 6-80 incoming\_connections

Field	Details
Description	This metric pegs active incoming connections from client to Ingress or Egress gateway.
Туре	Gauge
Dimension	Direction
	Host
	InstanceIdentifier

#### Table 6-81 outgoing\_connections

Field	Details
Description	This metric pegs active outgoing connections from Ingress gateway or Egress gateway to destination
Туре	Gauge
Dimension	<ul> <li>Direction</li> <li>Host</li> <li>InstanceIdentifier</li> </ul>

#### Table 6-82 sbitimer\_timezone\_mismatch

Field	Details
Description	This metric pegs when sbiTimerTimezone is set to ANY and time zone is not specified in the header then above metric is pegged in ingress and egress gateways.
Туре	Gauge
Dimension	<ul><li>Route_path</li><li>Method</li></ul>

#### Table 6-83 nrfclient\_nrf\_operative\_status

Field	Details
Description	The current operative status of the NRF Instance.  Note: The HealthCheck mechanism is an important component that allows monitoring and managing the health of NRF services.
	When enabled, it makes periodic HTTP requests to NRF services to check their availability and updates their status accordingly so that the metric nrfclient_nrf_operative_status updates properly.
	When disabled, for each NRF route, it is checked whether the retry time has expired. If so, the health state is reset to "HEALTHY", and the retry time is cleared.
Туре	Gauge
Dimension	NrfUri - URI of the NRF Instance

#### Table 6-84 nrfclient\_dns\_lookup\_request\_total

Field	Details
Description	Total number of times a DNS lookup request is sent to the alternate route service.
Туре	Counter



#### Table 6-84 (Cont.) nrfclient\_dns\_lookup\_request\_total

Field	Details
Dimension	Scheme
	VirtualFqdn

## 6.1.4 NSSF OAuth Metrics

This section provides details about the NSSF OAuth metrics.

Table 6-85 oc\_oauth\_nrf\_request\_total

Field	Details
Description	This is pegged in the OAuth client implementation if the request is sent to NRF for requesting the OAuth token. OAuth client implementation is used in Egress gateway.
Туре	Counter
Dimension	<ul> <li>ConsumerNFInstanceId</li> <li>ConsumerNFType</li> <li>TargetNFType TargetNFInstanceId</li> <li>scope NrfFqdn</li> </ul>

#### Table 6-86 oc\_oauth\_nrf\_response\_success\_total

Field	Details
Description	This is pegged in the OAuth client implementation if an OAuth token is successfully received from the NRF. OAuth client implementation is used in Egress gateway.
Туре	Counter
Dimension	<ul> <li>ConsumerNFInstanceId</li> <li>ConsumerNFType</li> <li>TargetNFType</li> <li>TargetNFInstanceId</li> <li>scope StatusCode NrfFqdn</li> </ul>

#### Table 6-87 oc\_oauth\_nrf\_response\_failure\_total

Field	Details
Description	This is pegged in the OAuthClientFilter in Egress gateway whenever GetAccessTokenFailedException is captured.
Туре	Counter
Dimension	<ul> <li>ConsumerNFInstanceId</li> <li>ConsumerNFType</li> <li>TargetNFType</li> <li>TargetNFInstanceId</li> <li>scope StatusCode NrfFqdn</li> </ul>



#### Table 6-88 oc\_oauth\_nrf\_response\_failure\_total

Field	Details
Description	This is pegged in the OAuthClientFilter in Egress gateway whenever GetAccessTokenFailedException is captured.
Туре	Counter
Dimension	<ul> <li>ConsumerNFInstanceId</li> <li>ConsumerNFType</li> <li>TargetNFType</li> <li>TargetNFInstanceId</li> <li>Scope</li> <li>StatusCode</li> <li>ErrorOriginator</li> <li>NrfFqdn</li> </ul>

#### Table 6-89 oc\_oauth\_request\_failed\_internal\_total

Field	Details
Description	This is pegged in the OAuthClientFilter in Egress gateway whenever InternalServerErrorException is captured.
Туре	Counter
Dimension	<ul> <li>ConsumerNFInstanceId</li> <li>ConsumerNFType</li> <li>TargetNFType</li> <li>TargetNFInstanceId</li> <li>scope</li> <li>StatusCode</li> <li>ErrorOriginator</li> <li>NrfFqdn</li> </ul>

#### Table 6-90 oc\_oauth\_token\_cache\_total

Field	Details
Description	This is pegged in the OAuth Client Implementation if the OAuth token is found in the cache.
Туре	Counter
Dimension	ConsumerNFInstanceId ConsumerNFType
	TargetNFType TargetNFInstanceId scope

#### Table 6-91 oc\_oauth\_request\_invalid\_total

Field	Details
Description	This is pegged in the OAuthClientFilter in Egress gateway whenever a BadAccessTokenRequestException/JsonProcessingException is captured.
Туре	Counter



#### Table 6-91 (Cont.) oc\_oauth\_request\_invalid\_total

Field	Details
Dimension	ConsumerNFInstanceId
	ConsumerNFType
	TargetNFType
	TargetNFInstanceId
	• scope
	StatusCode
	ErrorOriginator

#### Table 6-92 oc\_oauth\_validation\_successful\_total

Field	Details
Description	This is pegged in OAuth validator implementation if the received OAuth token is validated successfully. OAuth validator implementation is used in Ingress gateway.
Туре	Counter
Dimension	<ul><li>issuer</li><li>subject</li><li>scope</li></ul>

#### Table 6-93 oc\_oauth\_validation\_failure\_total

Field	Details
Description	This is pegged in OAuth validator implementation if the validation of the received OAuth token is failed. OAuth validator implementation is used in Ingress gateway.
Туре	Counter
Dimension	<ul><li>issuer</li><li>subject</li><li>scope</li><li>reason</li></ul>

#### Table 6-94 oc\_oauth\_cert\_expiryStatus

Field	Details
Description	Metric used to peg expiry date of the certificate. This metric is further used for raising alarms if certificate expires within 30 days or 7 days.
Туре	Gauge
Dimension	<ul><li>id</li><li>certificateName</li><li>secretName</li></ul>

#### Table 6-95 oc\_oauth\_cert\_loadStatus

Field	Details
Description	Metric used to peg whether given certificate can be loaded from secret or not. If it is loadable then "0" is pegged otherwise "1" is pegged. This metric is further used for raising alarms when certificate is not loadable.



#### Table 6-95 (Cont.) oc\_oauth\_cert\_loadStatus

Field	Details
Туре	Gauge
Dimension	<ul><li>id</li><li>certificateName</li><li>secretName</li></ul>

#### Table 6-96 oc\_oauth\_request\_failed\_cert\_expiry

Field	Details
Description	Metric used to keep track of number of requests with keyld in token that failed due to certificate expiry. Pegged whenever oAuth Validator module throws oauth custom exception due to certificate expiry for an incoming request.
Туре	Metric
Dimension	<ul> <li>target nf type</li> <li>target nf instance id</li> <li>consumer nf instance id</li> <li>nrf instance id</li> <li>service name of nf</li> <li>producer service key id</li> </ul>

#### Table 6-97 oc\_oauth\_keyid\_count

Field	Details
Description	Metric used to keep track of number of requests received with keyld in token. Pegged whenever a request with an access token containing kid in header comes to oAuth Validator. This is independent of whether the validation failed or was successful.
Туре	Metric
Dimension	<ul> <li>target nf type</li> <li>target nf instance id</li> <li>consumer nf instance id</li> <li>nrf instance id</li> <li>service name of nf</li> <li>producer service key id</li> </ul>

## 6.1.5 Managed Objects Metrics

This section provides details about the NSSF Managed Object (MO) metrics.

Table 6-98 ocnssf\_nssaiauth\_req\_rx

Field	Details
Description	Count of nssaiauth requests received by NSConfig.  Trigger Condition: Operator configuration of the Managed Object.
	Operator configuration of the Managed Object.
	This is pegged when HTTP GET, POST, DELETE, or PUT request is received by NSSF.
Туре	Counter



#### Table 6-98 (Cont.) ocnssf\_nssaiauth\_req\_rx

Field	Details
Service Operation	nssaiauth
Dimension	Method

#### Table 6-99 ocnssf\_nssaiauth\_res\_tx

Field	Details
Description	Count of successful responses sent by NSConfig for a nssaiauth request. Trigger Condition: Operator configuration of the Managed Object. This is pegged when a 2xx response for HTTP GET, POST, DELETE, or PUT request is sent by NSSF.
Туре	Counter
Service Operation	nssaiauth
Dimension	Method

#### Table 6-100 ocnssf\_nssaiauth\_error\_res\_tx

Field	Details
Description	Count of error responses sent by NSConfig for a nssaiauth request. Trigger Condition: Operator configuration of the Managed Object.
	This is pegged when non 2xx response for HTTP GET, POST, DELETE, or PUT request is sent by NSSF.
Туре	Counter
Service Operation	nssaiauth
Dimension	Method
	Status

#### Table 6-101 ocnssf\_nssaiauth\_created

Field	Details
Description	Count of nssaiauth created in the database.  Trigger Condition: Operator configuration of the Managed Object leading to storage of the Managed Object in the database and Autoconfiguration by learning from the AMF.
	This is pegged as source OperatorConfig when operator configuration is the source and pegged with LearnedConfigAMF when NsAvailabilityUpdate leads to storage of nssaiauth.
Туре	Counter
Service Operation	nssaiauth
Dimension	Source



#### Table 6-102 ocnssf\_nssaiauth\_deleted

Field	Details
Description	Count of nssaiauth deleted in the database.  Trigger Condition: Operator configuration of the Managed Object leading to deleting of the Managed Object in the database and autoconfiguration by learning from the AMF.
	This is pegged as source OperatorConfig when operator configuration is the source and pegged with LearnedConfigAMF when NSAvailability Update leads to storage of nssaiauth.
Туре	Counter
Service Operation	nssaiauth
Dimension	Source

#### Table 6-103 ocnssf\_nssaiauth\_updated

Field	Details
Description	Count of nssaiauth updated in the database.  Trigger Condition: Operator configuration of the Managed Object leading to deleting of the Managed Object in the database and autoconfiguration by learning from the AMF.
	This is pegged as source OperatorConfig when operator config is the source and pegged with LearnedConfigAMF when NSAvailability Update leads to storage of nssaiauth.
	<b>Note</b> : In current scenario, autoconfiguration does not update the Managed Object in the database, it only deletes and creates Managed Objects.
Туре	Counter
Service Operation	nssaiauth
Dimension	Source

#### Table 6-104 ocnssf\_nssaiauth\_error

Field	Details
Description	Count of failures on Managed Object processing. Trigger Condition: Error while creating, deleting, or updating a Managed object.
	This is pegged when error occurs while handling a Managed Object.
	Note: This must be pegged when ocnssf_nssaiauth_error_res_tx is pegged.
Туре	Counter
Service Operation	nssaiauth
Dimension	Source
	Operation
	• ERRORTYPE

#### Table 6-105 ocnssf\_nsiprofile\_req\_rx

Field	Details
Description	Count of nsiprofile requests received by NSConfig. Trigger Condition: Operator configuration of the Managed Object. Operator configuration of the Managed Object. This is pegged when HTTP GET, POST, DELETE, or PUT request is received by NSSF.
Туре	Counter



### Table 6-105 (Cont.) ocnssf\_nsiprofile\_req\_rx

Field	Details
Service Operation	nsiprofile
Dimension	Method

### Table 6-106 ocnssf\_amfset\_req\_rx

Field	Details
Description	Count of amfset requests received by NSConfig. Trigger Condition: Operator configuration of the Managed Object.
	Operator configuration of the Managed Object.
	This is pegged when HTTP GET, POST, DELETE, or PUT request is received by NSSF.
Туре	Counter
Service Operation	amfset
Dimension	Method

### Table 6-107 ocnssf\_amfset\_res\_tx

Field	Details
Description	Count of successful responses sent by NSConfig for a amfset request. Trigger Condition: Operator configuration of the Managed Object. This is pegged when a 2xx response for HTTP GET, POST, DELETE, or PUT request is sent by NSSF.
Туре	Counter
Service Operation	amfset
Dimension	Method

### Table 6-108 ocnssf\_amfset\_error\_res\_tx

Field	Details
Description	Count of error responses sent by NSConfig for a amfset request.  Trigger Condition: Operator configuration of the Managed Object.
	This is pegged when non 2xx response for HTTP GET, POST, DELETE, or PUT request is sent by NSSF.
Туре	Counter
Service Operation	amfset
Dimension	Method
	Status



### Table 6-109 ocnssf\_amfset\_created

Field	Details
Description	Count of amfset created in the database. Trigger Condition: Operator configuration of the Managed Object leading to storage of the Managed Object in the database and Autoconfiguration by learning from the AMF. This is pegged as source OperatorConfig when operator configuration is the source.
Туре	Counter
Service Operation	amfset
Dimension	Source

### Table 6-110 ocnssf\_amfset\_deleted

Field	Details
Description	Count of amfset deleted in the database.  Trigger Condition: Operator configuration of the Managed Object leading to deleting of the Managed Object in the database and autoconfiguration by learning from the AMF.  This is pegged as source OperatorConfig when operator configuration is the source.
Туре	Counter
Service Operation	amfset
Dimension	Source

### Table 6-111 ocnssf\_amfset\_updated

Field	Details
Description	Count of amfset updated in the database. Trigger Condition: Operator configuration of the Managed Object leading to deleting of the Managed Object in the database and autoconfiguration by learning from the AMF. This is pegged as source OperatorConfig when operator config is the source.
Туре	Counter
Service Operation	amfset
Dimension	Source

### Table 6-112 ocnssf\_amfset\_error

Field	Details
Description	Count of failures on Managed Object processing.  Trigger Condition: Error while creating, deleting, or updating a Managed object.  This is pegged when error occurs while handling a Managed Object.
Туре	Counter
Service Operation	amfset
Dimension	<ul><li>Source</li><li>Operation</li><li>ERRORTYPE</li></ul>



### Table 6-113 ocnssf\_amfresolution\_req\_rx

Field	Details
Description	Count of amfresolution requests received by NSConfig. Trigger Condition: Operator configuration of the Managed Object.
	Operator configuration of the Managed Object.
	This is pegged when HTTP GET, POST, DELETE, or PUT request is received by NSSF.
Туре	Counter
Service Operation	amfresolution
Dimension	Method

### Table 6-114 ocnssf\_amfresolution\_res\_tx

Field	Details
Description	Count of successful responses sent by NSConfig for a amfresolution request.  Trigger Condition: Operator configuration of the Managed Object.
	This is pegged when a 2xx response for HTTP GET, POST, DELETE, or PUT request is sent by NSSF.
Туре	Counter
Service Operation	amfresolution
Dimension	Method

### Table 6-115 ocnssf\_amfresolution\_error\_res\_tx

Field	Details
Description	Count of error responses sent by NSConfig for a amfresolution request. Trigger Condition: Operator configuration of the Managed Object.
	This is pegged when non 2xx response for HTTP GET, POST, DELETE, or PUT request is sent by NSSF.
Туре	Counter
Service Operation	amfresolution
Dimension	Method
	Status

### Table 6-116 ocnssf\_amfresolution\_created

Field	Details
Description	Count of amfresolution created in the database.  Trigger Condition: Operator configuration of the Managed Object leading to storage of the Managed Object in the database and Autoconfiguration by learning from the AMF.  This is pegged as source OperatorConfig when operator configuration is the source.
Туре	Counter
Service Operation	amfresolution
Dimension	Source



### Table 6-117 ocnssf\_amfresolution\_deleted

Field	Details
Description	Count of amfresolution deleted in the database.  Trigger Condition: Operator configuration of the Managed Object leading to deleting of the Managed Object in the database and autoconfiguration by learning from the AMF.  This is pegged as source OperatorConfig when operator configuration is the source.
Туре	Counter
Service Operation	amfresolution
Dimension	Source

### Table 6-118 ocnssf\_amfresolution\_updated

Field	Details
Description	Count of amfresolution updated in the database.  Trigger Condition: Operator configuration of the Managed Object leading to deleting of the Managed Object in the database and autoconfiguration by learning from the AMF.  This is pegged as source OperatorConfig when operator config is the source.
Туре	Counter
Service Operation	amfresolution
Dimension	Source

### Table 6-119 ocnssf\_amfresolution\_error

Field	Detaile
Field	Details
Description	Count of failures on Managed Object processing.  Trigger Condition: Error while creating, deleting, or updating a Managed object.  This is pegged when error occurs while handling a Managed Object.
	This is pegged when end occurs while handling a Managed Object.
Туре	Counter
Service Operation	amfresolution
Dimension	• Source
	<ul><li>Operation</li><li>ERRORTYPE</li></ul>

### Table 6-120 ocnssf\_timeprofile\_req\_rx

Field	Details
Description	Count oftimeprofile requests received by NSConfig. Trigger Condition: Operator configuration of the Managed Object.
	Operator configuration of the Managed Object.
	This is pegged when HTTP GET, POST, DELETE, or PUT request is received by NSSF.
Туре	Counter
Service Operation	timeprofile
Dimension	Method



### Table 6-121 ocnssf\_timeprofile\_res\_tx

Field	Details
Description	Count of successful responses sent by NSConfig for a timeprofile request.  Trigger Condition: Operator configuration of the Managed Object.  This is pegged when a 2xx response for HTTP GET, POST, DELETE, or PUT request is sent by NSSF.
Туре	Counter
Service Operation	timeprofile
Dimension	Method

### Table 6-122 ocnssf\_timeprofile\_error\_res\_tx

Field	Details
Description	Count of error responses sent by NSConfig for a timeprofile request.  Trigger Condition: Operator configuration of the Managed Object.
	This is pegged when non 2xx response for HTTP GET, POST, DELETE, or PUT request is sent by NSSF.
Туре	Counter
Service Operation	timeprofile
Dimension	Method
	Status

### Table 6-123 ocnssf\_timeprofile\_created

Field	Details
Description	Count of timeprofile created in the database.  Trigger Condition: Operator configuration of the Managed Object leading to storage of the Managed Object in the database and Autoconfiguration by learning from the AMF.  This is pegged as source OperatorConfig when operator configuration is the source.
Туре	Counter
Service Operation	timeprofile
Dimension	Source

### Table 6-124 ocnssf\_timeprofile\_deleted

Field	Details
Description	Count of timeprofile deleted in the database.  Trigger Condition: Operator configuration of the Managed Object leading to deleting of the Managed Object in the database and autoconfiguration by learning from the AMF.  This is pegged as source OperatorConfig when operator configuration is the source.
Туре	Counter
Service Operation	timeprofile
Dimension	Source



### Table 6-125 ocnssf\_timeprofile\_updated

Field	Details
Description	Count of timeprofile updated in the database.  Trigger Condition: Operator configuration of the Managed Object leading to deleting of the Managed Object in the database and autoconfiguration by learning from the AMF.  This is pegged as source OperatorConfig when operator config is the source.
Туре	Counter
Service Operation	timeprofile
Dimension	Source

### Table 6-126 ocnssf\_timeprofile\_error

Field	Details
Description	Count of failures on Managed Object processing.  Trigger Condition: Error while creating, deleting, or updating a Managed object.
	This is pegged when error occurs while handling a Managed Object.
Туре	Counter
Service Operation	timeprofile
Dimension	Source
	Operation
	ERRORTYPE

### Table 6-127 ocnssf\_defaultsnssai\_req\_rx

Field	Details
Description	Count of defaultsnssai requests received by NSConfig. Trigger Condition: Operator configuration of the Managed Object.
	Operator configuration of the Managed Object.
	This is pegged when HTTP GET, POST, DELETE, or PUT request is received by NSSF.
Туре	Counter
Service Operation	defaultsnssai
Dimension	Method

### Table 6-128 ocnssf\_defaultsnssai\_res\_tx

Field	Details
Description	Count of successful responses sent by NSConfig for a defaultsnssai request.  Trigger Condition: Operator configuration of the Managed Object.  This is pegged when a 2xx response for HTTP GET, POST, DELETE, or PUT request is sent by NSSF.
Туре	Counter
Service Operation	defaultsnssai
Dimension	Method



### Table 6-129 ocnssf\_defaultsnssai\_error\_res\_tx

Field	Details
Description	Count of error responses sent by NSConfig for a defaultsnssai request.  Trigger Condition: Operator configuration of the Managed Object.
	This is pegged when non 2xx response for HTTP GET, POST, DELETE, or PUT request is sent by NSSF.
Туре	Counter
Service Operation	defaultsnssai
Dimension	Method
	Status

### Table 6-130 ocnssf\_defaultsnssai\_created

Field	Details
Description	Count of defaultsnssai created in the database.  Trigger Condition: Operator configuration of the Managed Object leading to storage of the Managed Object in the database and Autoconfiguration by learning from the AMF.  This is pegged as source OperatorConfig when operator configuration is the source.
Туре	Counter
Service Operation	defaultsnssai
Dimension	Source

### Table 6-131 ocnssf\_defaultsnssai\_deleted

Field	Details
Description	Count of defaultsnssai deleted in the database.  Trigger Condition: Operator configuration of the Managed Object leading to deleting of the Managed Object in the database and autoconfiguration by learning from the AMF.  This is pegged as source OperatorConfig when operator configuration is the source.
Туре	Counter
Service Operation	defaultsnssai
Dimension	Source

### Table 6-132 ocnssf\_defaultsnssai\_updated

Field	Details
Description	Count of defaultsnssai updated in the database. Trigger Condition: Operator configuration of the Managed Object leading to deleting of the Managed Object in the database and autoconfiguration by learning from the AMF. This is pegged as source OperatorConfig when operator config is the source.
Туре	Counter
Service Operation	defaultsnssai
Dimension	Source



### Table 6-133 ocnssf\_mappingofnssai\_req\_rx

Field	Details
Description	Count of mappingofnssai requests received by NSConfig. Trigger Condition: Operator configuration of the Managed Object.
	Operator configuration of the Managed Object.
	This is pegged when HTTP GET, POST, DELETE, or PUT request is received by NSSF.
Туре	Counter
Service Operation	mappingofnssai
Dimension	Method

### Table 6-134 ocnssf\_mappingofnssai\_res\_tx

Field	Details
Description	Count of successful responses sent by NSConfig for a mappingofnssai request.  Trigger Condition: Operator configuration of the Managed Object.
	This is pegged when a 2xx response for HTTP GET, POST, DELETE, or PUT request is sent by NSSF.
Туре	Counter
Service Operation	mappingofnssai
Dimension	Method

### Table 6-135 ocnssf\_mappingofnssai\_error\_res\_tx

Field	Details
Description	Count of error responses sent by NSConfig for a mappingofnssai request. Trigger Condition: Operator configuration of the Managed Object.
	This is pegged when non 2xx response for HTTP GET, POST, DELETE, or PUT request is sent by NSSF.
Туре	Counter
Service Operation	mappingofnssai
Dimension	Method
	Status

### Table 6-136 ocnssf\_mappingofnssai\_created

Field	Details
Description	Count of mappingofnssai created in the database.  Trigger Condition: Operator configuration of the Managed Object leading to storage of the Managed Object in the database and Autoconfiguration by learning from the AMF.  This is pegged as source OperatorConfig when operator configuration is the source.
Туре	Counter
Service Operation	mappingofnssai
Dimension	Source



Table 6-137 ocnssf\_mappingofnssai\_deleted

Field	Details
Description	Count of mappingofnssai deleted in the database.  Trigger Condition: Operator configuration of the Managed Object leading to deleting of the Managed Object in the database and autoconfiguration by learning from the AMF.  This is pegged as source OperatorConfig when operator configuration is the source.
Туре	Counter
Service Operation	mappingofnssai
Dimension	Source

Table 6-138 ocnssf\_mappingofnssai\_updated

Field	Details
Description	Count of mappingofnssai updated in the database. Trigger Condition: Operator configuration of the Managed Object leading to deleting of the Managed Object in the database and autoconfiguration by learning from the AMF. This is pegged as source OperatorConfig when operator config is the source.
Туре	Counter
Service Operation	mappingofnssai
Dimension	Source

### 6.1.6 Perf-info metrics for Overload Control

This section provides details about Perf-info metrics for overload control.

Table 6-139 cgroup\_cpu\_nanoseconds

Field	Details
Description	Reports the total CPU time (in nanoseconds) on each CPU core for all the tasks in the cgroup.
Туре	Gauge
Dimension	NA

#### Table 6-140 cgroup\_memory\_bytes

Field	Details
Description	Reports the memory usage.
Туре	Gauge
Dimension	NA

### Table 6-141 load\_level

Field	Details
Description	Provides information about the overload manager load level.
Туре	Gauge



### Table 6-141 (Cont.) load\_level

Field	Details
Dimension	service
	namespace

# 6.1.7 Egress Gateway Metrics

This section provides details about Egress Gateway metrics.

Table 6-142 oc\_egressgateway\_outgoing\_tls\_connections

Field	Details
Description	Number of TLS connections received on the Egress Gateway and their negotiated TLS versions. The versions can be TLSv1.3 or TLSv1.2
Туре	Gauge
Service Operation	Egress Gateway
Dimension	<ul> <li>NegotiatedTLSVersion</li> <li>Host</li> <li>Direction</li> <li>InstanceIdentifier</li> </ul>

### Table 6-143 oc\_fqdn\_alternate\_route\_total

Field	Details
Description	Tracks number of registration, deregistration and GET calls received for a given scheme and FQDN.  Note: Registration does not reflect active registration numbers. It captured number of registration requests received.
Туре	Counter
Service Operation	Egress Gateway
Dimension	<ul><li>type: Register/Deregister/GET</li><li>binding_value: <scheme>+<fqdn></fqdn></scheme></li></ul>

### Table 6-144 oc\_dns\_srv\_lookup\_total

Field	Details
Description	Track number of time DNS SRV lookup was done for a given scheme and FQDN.
Туре	Counter
Service Operation	Egress Gateway
Dimension	binding_value: <scheme>+<fqdn></fqdn></scheme>



### Table 6-145 oc\_alternate\_route\_resultset

Field	Details
Description	Value provides number of alternate routes known for a given scheme and FQDN.  Whenever DNS SRV lookup or static configuration is done, this metric provide number of known alternate route for a given pair. For example, <"http", "abc.oracle.com">: 2.
Туре	Gauge
Service Operation	Egress Gateway
Dimension	binding_value: <scheme>+<fqdn></fqdn></scheme>

### Table 6-146 oc\_configclient\_request\_total

Field	Details
Description	This metric is pegged whenever a polling request is made from config client to the server for configuration updates.
Туре	Counter
Service Operation	Egress Gateway
Dimension	<ul> <li>Tags: releaseVersion, configVersion.</li> <li>releaseVersion tag indicates the current chart version of alternate route service deployed.</li> <li>configVersion tag indicates the current configuration version of alternate route service.</li> </ul>

### Table 6-147 oc\_configclient\_response\_total

Field	Details
Description	This metric is pegged whenever a response is received from the server to client.
Туре	Counter
Service Operation	Egress Gateway
Dimension	Tags: releaseVersion, configVersion, updated.
	<ul> <li>releaseVersion tag indicates the current chart version of alternate route service deployed.</li> <li>configVersion tag indicates the current configuration version of alternate route service.</li> </ul>
	<ul> <li>updated tag indicates whether there is a configuration update or not.</li> </ul>

### Table 6-148 oc\_egressgateway\_peer\_health\_status

Field	Details
Description	It defines Egress Gateway peer health status. This metric is set to 1, if a peer is unhealthy.
	This metric is reset to 0, when it becomes healthy again.
Туре	Gauge
Service Operation	Egress Gateway
Dimension	<ul><li>peer</li><li>vfqdn</li></ul>



### Table 6-149 oc\_egressgateway\_peer\_health\_ping\_request\_total

Field	Details
Description	It defines Egress Gateway peer health ping request. This metric is incremented every time Egress Gateway send a health ping towards a peer.
Туре	Counter
Service Operation	Egress Gateway
Dimension	<ul><li>peer</li><li>vfqdn</li><li>statusCode</li><li>cause</li></ul>

### Table 6-150 oc\_egressgateway\_peer\_health\_ping\_response\_total

Field	Details
Description	Egress Gateway Peer health ping response. This metric is incremented every time a Egress Gateway receives a health ping response (irrespective of success or failure) from a peer.
Туре	Counter
Service Operation	Egress Gateway
Dimension	<ul><li>peer</li><li>vfqdn</li><li>statusCode</li><li>cause</li></ul>

### Table 6-151 oc\_egressgateway\_peer\_health\_status\_transitions\_total

Field	Details
Description	It defines Egress Gateway peer health status transitions. Egress Gateway increments this metric every time a peer transitions from available to unavailable or unavailable to available.
Туре	Counter
Service Operation	Egress Gateway
Dimension	<ul><li>peer</li><li>vfqdn</li><li>from</li><li>to</li></ul>

### Table 6-152 oc\_egressgateway\_peer\_count

Field	Details
Description	It defines Egress Gateway peer count. This metric is incremented every time for the peer count.
Туре	Gauge
Service Operation	Egress Gateway
Dimension	peerset



Table 6-153 oc\_egressgateway\_peer\_available\_count

Field	Details
Description	It defines Egress Gateway available peer count. This metric is incremented every time for the available peer count.
Туре	Gauge
Service Operation	Egress Gateway
Dimension	peerset

Table 6-154 oc\_egressgateway\_user\_agent\_consumer

Field	Details
Description	Whenever the feature is enabled and User-Agent Header is getting generated.
Туре	Counter
Service Operation	Egress Gateway
Dimension	ConsumerNfInstanceId: ID of consumer NF (NSSF) as configured in Egress Gateway.

# 6.1.8 Ingress Gateway Metrics

This section provides details about Ingress Gateway metrics.

Table 6-155 oc\_ingressgateway\_incoming\_tls\_connections

Field	Details
Description	Number of TLS connections received on the Ingress Gateway and their negotiated TLS versions. The versions can be TLSv1.3 or TLSv1.2.
Туре	Gauge
Service Operation	Ingress Gateway
Dimension	<ul> <li>NegotiatedTLSVersion</li> <li>Host</li> <li>Direction</li> <li>InstanceIdentifier</li> </ul>

Table 6-156 oc\_ingressgateway\_pod\_congestion\_state

Field	Details
Description	It is used to track congestion state of a pod.
Туре	Gauge
Service Operation	Ingress Gateway
Dimension	level = 0,1,2



### Table 6-157 oc\_ingressgateway\_pod\_resource\_stress

Field	Details
Description	It tracks CPU, memory, and queue usage (as percentages) to determine the congestion state of the POD that is performing the calculations.
Туре	Gauge
Service Operation	Ingress Gateway
Dimension	type = "PendingRequest","CPU","Memory"

#### Table 6-158 oc\_ingressgateway\_pod\_resource\_state

Field	Details
Description	It tracks the congestion state of individual resources, which is calculated based on their usage and the configured threshold.
Туре	Gauge
Service Operation	Ingress Gateway
Dimension	type = "PendingRequest","CPU","Memory" level = 0,1,2
	0: Normal
	• 1: DOC
	2: Congested

Table 6-159 oc\_ingressgateway\_incoming\_pod\_connections\_rejected\_total

Field	Details
Description	It tracks the number of connections dropped in the congested or Danger Of Congestion (DOC) state.
Туре	Counter
Service Operation	Ingress Gateway
Dimension	NA

## 6.2 NSSF KPIs

This section includes information about KPIs for Oracle Communications Cloud Native Core, Network Slice Selection Function.

The following are the NSSF KPIs:

## 6.2.1 NSSelection KPIs

Table 6-160 NSSF NSSelection Initial Registration Success Rate

Field	Details
Description	Percentage of NSSelection Initial registration messages with success response
Expression	sum(ocnssf_nsselection_success_tx_total{message_type=\"registration\"})/ sum(ocnssf_nsselection_rx_total{message_type=\"registration\"}))*100"



### Table 6-161 NSSF NSSelection PDU establishment success rate

Field	Details
Description	Percentage of NSSelection PDU establishment messages with success response
Expression	sum(ocnssf_nsselection_success_tx_total{message_type=\"pdu_session\"})/ sum(ocnssf_nsselection_rx_total{message_type=\"pdu_session\"}))*100"

### Table 6-162 NSSF NSSelection UE-Config Update success rate

Field	Details
Description	Percentage of NSSelection UE-Config Update messages with success response
Expression	sum(ocnssf_nsselection_success_tx_total{message_type=\"ue_config_update\"})/ sum(ocnssf_nsselection_rx_total{message_type=\"ue_config_update\"}))*100",

### Table 6-163 4xx Responses (NSSelection)

Field	Details
Description	Rate of 4xx response for NSSelection
Expression	sum(increase(oc_ingressgateway_http_responses{Status=~"4.* ",Uri=~".*nnssf-nsselection.*",Method="GET"}[5m]))

### Table 6-164 5xx Responses (NSSelection)

Field	Details	
Description	Rate of 5xx response for NSSelection	
Expression	sum(increase(oc_ingressgateway_http_responses{Status=~"5.* ",Uri=~".*nnssf-nsselection.*",Method="GET"}[5m])	

# 6.2.2 NSAvailability KPIs

### Table 6-165 NSSF NSAvailability PUT success rate

Field	Details	
Description	ercentage of NSAvailability UPDATE PUT messages with success response	
Expression	sum(ocnssf_nssaiavailability_success_tx_total{message_type=\"availability_update\"} {method=\"PUT"})/sum(ocnssf_nssaiavailability_rx_total{message_type=\"availability_update\"} {method=\"PUT"}))*100"	

### Table 6-166 NSSF NSAvailability PATCH success rate

Field	Details	
Description	ercentage of NSAvailability UPDATE PATCH messages with success response	
Expression	sum(ocnssf_nssaiavailability_success_tx_total{message_type=\"availability_update\"} {method=\"PATCH"})/sum(ocnssf_nssaiavailability_rx_total{message_type=\"availability_update\"} {method=\"PATCH"}))*100"	



### Table 6-167 NSSF NSAvailability Delete success rate

Field	Details
Description	Percentage of NSAvailability Delete messages with success response
Expression	sum(ocnssf_nssaiavailability_success_tx_total{message_type=\"availability_update\"} {method=\"DELETE"})/ sum(ocnssf_nssaiavailability_rx_total{message_type=\"availability_update\"} {method=\"DELETE"}))*100""

### Table 6-168 NSSF NSAvailability Subscribe success rate

Field	Details
Description	Percentage of NSAvailability Subscribe messages with success response
Expression	sum(ocnssf_nssaiavailability_success_tx_total{message_type=\"availability_subscribe\"} {method=\"POST"})/ sum(ocnssf_nssaiavailability_rx_total{message_type=\"availability_subscribe\"} {method=\"POST"}))*100"

### Table 6-169 NSSF NSAvailability Unsubscribe success rate

Field	Details
Description	Percentage of NSAvailability Unsubscribe messages with success response
Expression	sum(ocnssf_nssaiavailability_success_tx_total{message_type=\"availability_subscribe\"} {method=\"DELETE"})/ sum(ocnssf_nssaiavailability_rx_total{message_type=\"availability_subscribe\"} {method=\"DELETE"}))*100"

### Table 6-170 4xx Responses (NSAvailability)

Field	Details	
Description	Rate of 4xx response for NSAvailability	
Expression	sum(increase(oc_ingressgateway_http_responses{Status=~"4.* ",Uri=~".*nnssf-nsavailability.*",Method="GET"}[5m]))	

### Table 6-171 5xx Responses (NSAvailability)

Field	Details	
Description	Rate of 5xx response for NSAvailability	
Expression	sum(increase(oc_ingressgateway_http_responses{Status=~"4.* ",Uri=~".*nnssf-nsavailability.*",Method="GET"}[5m]))	

# 6.2.3 Ingress Gateway KPIs

### **Table 6-172 NSSF Ingress Request**

Field	Details	
Description	Rate of HTTP requests received at NSSF Ingress Gateway	



### Table 6-172 (Cont.) NSSF Ingress Request

Field	Details	
Expression	oc_ingressgateway_http_requests	

### 6.3 NSSF Alerts

This section includes information about alerts for Oracle Communications Network Slice Selection Function.



#### (i) Note

The performance and capacity of the NSSF system may vary based on the call model, feature or interface configuration, network conditions, and underlying CNE and hardware environment.

You can configure alerts in Prometheus and ocnssf\_alert\_rules\_25.1.100.yaml file.

The following table describes the various severity types of alerts generated by NSSF:

#### Table 6-173 Alerts Levels or Severity Types

Alerts Levels / Severity Types	Definition	
Critical	Indicates a severe issue that poses a significant risk to safety, security, or operational integrity. It requires immediate response to address the situation and prevent serious consequences. Raised for conditions may affect the service of NSSF.	
Major	Indicates a more significant issue that has an impact on operations or poses a moderate risk. It requires prompt attention and action to mitigate potential escalation. Raised for conditions may affect the service of NSSF.	
Minor	Indicates a situation that is low in severity and does not pose an immediate risk to safety, security, or operations. It requires attention but does not demand urgent action. Raised for conditions may affect the service of NSSF.	
Info or Warn (Informational)	Provides general information or updates that are not related to immediate risks or actions. These alerts are for awareness and do not typically require any specific response. WARN and INFO alerts may not impact the service of NSSF.	



### **⚠** Caution

User, computer and applications, and character encoding settings may cause an issue when copy-pasting commands or any content from PDF. The PDF reader version also affects the copy-pasting functionality. It is recommended to verify the pasted content when the hyphens or any special characters are part of the copied content.



#### (i) Note

- kubect1 commands might vary based on the platform deployment. Replace kubect1 with Kubernetes environment-specific command line tool to configure Kubernetes resources through kube-api server. The instructions provided in this document are as per the Oracle Communications Cloud Native Environment (OCCNE) version of kube-api server.
- The alert file can be customized as required by the deployment environment. For example, namespace can be added as a filtered criteria to the alert expression to filter alerts only for a specific namespace.

## 6.3.1 System Level Alerts

This section lists the system level alerts.

### 6.3.1.1 OcnssfNfStatusUnavailable

Table 6-174 OcnssfNfStatusUnavailable

Field	Details
Description	'OCNSSF services unavailable'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : All OCNSSF services are unavailable.'
Severity	Critical
Condition	All the NSSF services are unavailable, either because the NSSF is getting deployed or purged. These NSSF services considered are nssfselection, nssfsubscription, nssfavailability, nssfconfiguration, appinfo, ingressgateway and egressgateway.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9001
Metric Used	'up'
	<b>Note:</b> This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.



Table 6-174 (Cont.) OcnssfNfStatusUnavailable

Field	Details
Recommended Actions	The alert is cleared automatically when the NSSF services start becoming available. <b>Steps:</b>
	<ol> <li>Check for service specific alerts which may be causing the issues with service exposure.</li> </ol>
	2. Run the following command to check if the pod's status is in "Running" state:
	kubectl -n <namespace> get pod</namespace>
	If it is not in running state, capture the pod logs and events.
	Run the following command to fetch the events as follows:
	<pre>kubectl get eventssort-by=.metadata.creationTimestamp -n <namespace></namespace></pre>
	3. Refer to the application logs on Kibana and check for database related failures such as connectivity, invalid secrets, and so on. The logs can be filtered based on the services.
	4. Run the following command to check Helm status and make sure there are no errors:
	helm status <helm desired="" name="" nf="" of="" release="" the=""> -n <namespace></namespace></helm>
	If it is not in "STATUS: DEPLOYED", then again capture logs and events.
	5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.
	<b>Note:</b> Use Cloud Native Core Network Function Data Collector tool for capturing the logs. For more information, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.</i>

### 6.3.1.2 OcnssfPodsRestart

Table 6-175 OcnssfPodsRestart

Field	Details
Description	'Pod <pod name=""> has restarted.</pod>
Summary	'kubernetes_namespace: {{\$labels.namespace}}, podname: {{\$labels.pod}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : A Pod has restarted'
Severity	Major
Condition	A pod belonging to any of the NSSF services has restarted.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9002
Metric Used	'kube_pod_container_status_restarts_total'Note: This is a Kubernetes metric. If this metric is not available, use the similar metric as exposed by the monitoring system.



Table 6-175 (Cont.) OcnssfPodsRestart

Field	Details
Recommended Actions	The alert is cleared automatically if the specific pod is up.  Steps:
	<ol> <li>Refer to the application logs on Kibana and filter based on the pod name. Check for database related failures such as connectivity, Kubernetes secrets, and so on.</li> </ol>
	2. Run the following command to check orchestration logs for liveness or readiness probe failures:
	kubectl get po -n <namespace></namespace>
	Note the full name of the pod that is not running, and use it in the following command:
	kubectl describe pod <desired full="" name="" pod=""> -n <namespace></namespace></desired>
	3. Check the database status. For more information, see "Oracle Communications Cloud Native Core, cnDBTier User Guide".
	<ol> <li>If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.</li> <li>Note: Use Cloud Native Core Network Function Data Collector tool for capturing the logs. For more information, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.</li> </ol>

# 6.3.1.3 OcnssfSubscriptionServiceDown

Table 6-176 OcnssfSubscriptionServiceDown

Field	Details
Description	'OCNSSF Subscription service <ocnssf-nssubscription> is down'</ocnssf-nssubscription>
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : NssfSubscriptionServiceDown service down'
Severity	Critical
Condition	NssfSubscription services is unavailable.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9003
Metric Used	"up'
	<b>Note:</b> This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.



Table 6-176 (Cont.) OcnssfSubscriptionServiceDown

Field	Details
Recommended Actions	The alert is cleared when the NssfSubscription services is available.  Steps:
	<ol> <li>Check if NfService specific alerts are generated to understand which service is down.</li> <li>If the following alerts are generated based on which service is down</li> </ol>
	OcnssfSubscriptionServiceDown
	2. Run the following command to check the orchestration log nfsubscription service and check for liveness or readiness probe failures:
	kubectl get po -n <namespace></namespace>
	Note the full name of the pod that is not running, and use it in the following command:
	<pre>kubectl describe pod <specific desired="" full="" name="" pod=""> -n <namespace></namespace></specific></pre>
	3. Run the following command to check if the pod's status is in "Running" state:
	kubectl -n <namespace> get pod</namespace>
	If it is not in running state, capture the pod logs and events . Run the following command to fetch events:
	<pre>kubectl get eventssort-by=.metadata.creationTimestamp -n <namespace></namespace></pre>
	<ol> <li>Refer to the application logs on Kibana and filter based on above service names. Check for ERROR WARNING logs for each of these services.</li> </ol>
	5. Check the database status. For more information, see "Oracle Communications Cloud Native Core, cnDBTier User Guide".
	<ol><li>Refer to the application logs on Kibana and check for the service status of the nssfConfig service.</li></ol>
	<ol> <li>If the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</li> <li>Note: Use Cloud Native Core Network Function Data Collector tool for capturing the logs. For more information, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.</li> </ol>

### 6.3.1.4 OcnssfSelectionServiceDown

### Table 6-177 OcnssfSelectionServiceDown

Field	Details
Description	'OCNSSF Selection service <ocnssf-nsselection> is down'.</ocnssf-nsselection>



Table 6-177 (Cont.) OcnssfSelectionServiceDown

Field	Details
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : OcnssfSelectionServiceDown service down'
Severity	Critical
Condition	None of the pods of the NSSFSelection microservice is available.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9004
Metric Used	'up'  Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.
Recommended Actions	The alert is cleared when the nfsubscription service is available.  Steps:
	Run the following command to check the orchestration logs of ocnssf-nsselection service and check for liveness or readiness probe failures:
	kubectl get po -n <namespace></namespace>
	Note the full name of the pod that is not running, and use it in the following command:
	kubectl describe pod <specific desired="" full="" name="" pod=""> -n <namespace></namespace></specific>
	Refer to the application logs on Kibana and filter based on ocnssf-nsselection service names. Check for ERROR WARNING logs.
	3. Check the database status. For more information, see "Oracle Communications Cloud Native Core, cnDBTier User Guide".
	4. Depending on the failure reason, take the resolution steps.
	<ol> <li>If the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</li> <li>Note: Use Cloud Native Core Network Function Data Collector tool for capturing the logs. For more information, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.</li> </ol>

# 6.3.1.5 OcnssfAvailabilityServiceDown

Table 6-178 OcnssfAvailabilityServiceDown

Field	Details
Description	'Ocnssf Availability service ocnssf-nsavailability is down'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : NssfAvailability service down'
Severity	Critical
Condition	None of the pods of the OcnssfAvailabilityServiceDown microservice is available.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9005



Table 6-178 (Cont.) OcnssfAvailabilityServiceDown

Field	Details
Metric Used	'up'  Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.
Recommended Actions	The alert is cleared when the ocnssf-nsavailability service is available.  Steps:
	Run the following command to check the orchestration logs of ocnssf- nsavailability service and check for liveness or readiness probe failures:
	kubectl get po -n <namespace></namespace>
	Note the full name of the pod that is not running, and use it in the following command:
	kubectl describe pod <specific desired="" full="" name="" pod=""> -n <namespace></namespace></specific>
	Refer to the application logs on Kibana and filter based on ocnssf-nsavailability service names. Check for ERROR WARNING logs.
	3. Check the database status. For more information, see "Oracle Communications Cloud Native Core, cnDBTier User Guide".
	4. Depending on the failure reason, take the resolution steps.
	<ol> <li>If the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</li> <li>Note: Use Cloud Native Core Network Function Data Collector tool for capturing the logs. For more information, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.</li> </ol>

# 6.3.1.6 OcnssfConfigurationServiceDown

Table 6-179 OcnssfConfigurationServiceDown

Field	Details
Description	'OCNSSF Config service nssfconfiguration is down'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : OcnssfConfigServiceDown service down'
Severity	Critical
Condition	None of the pods of the NssfConfiguration microservice is available.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9006
Metric Used	'up'
	<b>Note:</b> : This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.



Table 6-179 (Cont.) OcnssfConfigurationServiceDown

Field	Details
Recommended Actions	The alert is cleared when the nssfconfiguration service is available.
	Steps:
	1. Run the following command to check the orchestration logs of nssfconfiguration service and check for liveness or readiness probe failures:
	kubectl get po -n <namespace></namespace>
	Note the full name of the pod that is not running, and use in the following command:
	<pre>kubectl describe pod <specific desired="" full="" name="" pod=""> -n <namespace></namespace></specific></pre>
	<ol><li>Refer the application logs on Kibana and filter based on nssfconfiguration service names. Check for ERROR WARNING logs related to thread exceptions.</li></ol>
	3. Check the database status. For more information, see "Oracle Communications Cloud Native Core, cnDBTier User Guide".
	4. Depending on the reason of failure, take the resolution steps.
	<ol> <li>If the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</li> <li>Note: Use Cloud Native Core Network Function Data Collector tool for capturing the logs. For more information, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.</li> </ol>

# 6.3.1.7 OcnssfAppInfoServiceDown

Table 6-180 OcnssfAppInfoServiceDown

Field	Details
Description	OCNSSF Appinfo service appinfo is down'
Summary	kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : Appinfo service down'
Severity	Critical
Condition	None of the pods of the App Info microservice is available.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9007
Metric Used	'up'
	<b>Note:</b> This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.



Table 6-180 (Cont.) OcnssfAppInfoServiceDown

Field	Details
Recommended Actions	The alert is cleared when the app-info service is available.
	Steps:
	Run the following command to check the orchestration logs of appinfo service and check for liveness or readiness probe failures:
	kubectl get po -n <namespace></namespace>
	Note the full name of the pod that is not running, and use it in the following command:
	kubectl describe pod <specific desired="" full="" name="" pod=""> -n <namespace></namespace></specific>
	<ol> <li>Refer to the application logs on Kibana and filter based on appinfo service names. Check for ERROR WARNING logs related to thread exceptions.</li> </ol>
	3. Depending on the failure reason, take the resolution steps.
	<ol> <li>If the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</li> <li>Note: Use Cloud Native Core Network Function Data Collector tool for capturing the logs. For more information, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.</li> </ol>

# 6.3.1.8 OcnssfIngressGatewayServiceDown

Table 6-181 OcnssflngressGatewayServiceDown

Field	Details
Description	'Ocnssf Ingress-Gateway service ingressgateway is down'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : OcnssflngressGwServiceDown service down'
Severity	Critical
Condition	None of the pods of the Ingress-Gateway microservice is available.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9008
Metric Used	'up'
	<b>Note:</b> This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.



Table 6-181 (Cont.) OcnssflngressGatewayServiceDown

Field	Details
Recommended Actions	The alert is cleared when the ingressgateway service is available.
	Steps:
	Run the following command to check the orchestration logs of ingress-gateway service and check for liveness or readiness probe failures:
	kubectl get po -n <namespace></namespace>
	Note the full name of the pod that is not running, and use it in the following command:
	kubectl describe pod <specific desired="" full="" name="" pod=""> -n <namespace></namespace></specific>
	Refer to the application logs on Kibana and filter based on ingress-gateway service names. Check for ERROR WARNING logs related to thread exceptions.
	3. Depending on the failure reason, take the resolution steps.
	<ol> <li>If the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</li> <li>Note: Use Cloud Native Core Network Function Data Collector tool for capturing the logs. For more information, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.</li> </ol>

# 6.3.1.9 OcnssfEgressGatewayServiceDown

Table 6-182 OcnssfEgressGatewayServiceDown

Field	Details
Description	'OCNSSF Egress service egressgateway is down'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : OcnssfEgressGwServiceDown service down'
Severity	Critical
Condition	None of the pods of the Egress-Gateway microservice is available.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9009
Metric Used	'up'
	<b>Note:</b> This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.



Table 6-182 (Cont.) OcnssfEgressGatewayServiceDown

Field	Details
Recommended Actions	The alert is cleared when the egressgateway service is available.
	Note: The threshold is configurable in the alerts.yaml
	Steps:
	Run the following command to check the orchestration logs of egress-gateway service and check for liveness or readiness probe failures:
	kubectl get po -n <namespace></namespace>
	Note the full name of the pod that is not running, and use it in the following command:
	kubectl describe pod <specific desired="" full="" name="" pod=""> -n <namespace></namespace></specific>
	Refer to the application logs on Kibana and filter based on egress-gateway service names. Check for ERROR WARNING logs related to thread exceptions.
	3. Depending on the failure reason, take the resolution steps.
	<ol> <li>If the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</li> <li>Note: Use Cloud Native Core Network Function Data Collector tool for capturing the logs. For more information, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.</li> </ol>

# 6.3.1.10 OcnssfOcpmConfigServiceDown

Table 6-183 OcnssfOcpmConfigServiceDown

Field	Details
Description	'OCNSSF OCPM Config service is down'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : Ocnssf OCPM Config service down'
Severity	Critical
Condition	None of the pods of the ConfigService is available.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9037
Metric Used	'up'
	<b>Note:</b> This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.



Table 6-183 (Cont.) OcnssfOcpmConfigServiceDown

Field	Details
Recommended Actions	The alert is cleared when the ConfigService is available.  Note: The threshold is configurable in the alerts.yaml  Steps:
	Run the following command to check the orchestration logs of ConfigService service and check for liveness or readiness probe failures:
	kubectl get po -n <namespace></namespace>
	Note the full name of the pod that is not running, and use it in the following command:
	kubectl describe pod <specific desired="" full="" name="" pod=""> -n <namespace></namespace></specific>
	Refer to the application logs on Kibana and filter based on PerfInfo service names. Check for ERROR WARNING logs related to thread exceptions.
	3. Depending on the failure reason, take the resolution steps.
	<ol> <li>If the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</li> <li>Note: Use Cloud Native Core Network Function Data Collector tool for capturing the logs. For more information, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.</li> </ol>

## 6.3.1.11 OcnssfPerfInfoServiceDown

Table 6-184 OcnssfPerfInfoServiceDown

Field	Details
Description	OCNSSF PerfInfo service is down'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : Ocnssf PerfInfo service down'
Severity	Critical
Condition	None of the pods of the PerfInfo service is available.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9036
Metric Used	'up'
	<b>Note:</b> This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.



Table 6-184 (Cont.) OcnssfPerfInfoServiceDown

Field	Details
Recommended Actions	The alert is cleared when the PerfInfo service is available.
	Note: The threshold is configurable in the alerts.yaml
	Steps:
	<ol> <li>Run the following command to check the orchestration logs of PerfInfo service and check for liveness or readiness probe failures:</li> </ol>
	kubectl get po -n <namespace></namespace>
	Note the full name of the pod that is not running, and use it in the following command:
	kubectl describe pod <specific desired="" full="" name="" pod=""> -n <namespace></namespace></specific>
	<ol><li>Refer to the application logs on Kibana and filter based on PerfInfo service names. Check for ERROR WARNING logs related to thread exceptions.</li></ol>
	3. Depending on the failure reason, take the resolution steps.
	<ol> <li>If the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</li> <li>Note: Use Cloud Native Core Network Function Data Collector tool for capturing the logs. For more information, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.</li> </ol>

# 6.3.1.12 OcnssfNrfClientManagementServiceDown

Table 6-185 OcnssfNrfClientManagementServiceDown

Field	Details
Description	'OCNSSF NrfClient Management service is down'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : Ocnssf NrfClient Management service down'
Severity	Critical
Condition	None of the pods of the NrfClientManagement service is available.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9034
Metric Used	'up'
	<b>Note:</b> This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.



Table 6-185 (Cont.) OcnssfNrfClientManagementServiceDown

Field	Details
Recommended Actions	The alert is cleared when the NrfClientManagement service is available.
	Note: The threshold is configurable in the alerts.yaml
	Steps:
	Run the following command to check the orchestration logs of     NrfClientManagement service and check for liveness or readiness probe failures:
	kubectl get po -n <namespace></namespace>
	Note the full name of the pod that is not running, and use it in the following command:
	kubectl describe pod <specific desired="" full="" name="" pod=""> -n <namespace></namespace></specific>
	2. Refer to the application logs on Kibana and filter based on NrfClientManagement service names. Check for ERROR WARNING logs related to thread exceptions.
	3. Depending on the failure reason, take the resolution steps.
	<ol> <li>If the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</li> <li>Note: Use Cloud Native Core Network Function Data Collector tool for capturing the logs. For more information, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.</li> </ol>

# 6.3.1.13 OcnssfNrfClientDiscoveryServiceDown

Table 6-186 OcnssfNrfClientDiscoveryServiceDown

Field	Details
Description	'OCNSSF NrfClient Discovery service is down'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : Ocnssf NrfClient Discovery service down'
Severity	Critical
Condition	None of the pods of the NrfClient Discovery service is available.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9033
Metric Used	'up'
	<b>Note:</b> This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.



Table 6-186 (Cont.) OcnssfNrfClientDiscoveryServiceDown

Field	Details
Recommended Actions	The alert is cleared when the NrfClient Discovery service is available.  Note: The threshold is configurable in the alerts.yaml  Steps:
	Run the following command to check the orchestration logs of NrfClient Discovery service and check for liveness or readiness probe failures:
	kubectl get po -n <namespace></namespace>
	Note the full name of the pod that is not running, and use it in the following command:
	kubectl describe pod <specific desired="" full="" name="" pod=""> -n <namespace></namespace></specific>
	Refer to the application logs on Kibana and filter based on NrfClient Discovery service names. Check for ERROR WARNING logs related to thread exceptions.
	3. Depending on the failure reason, take the resolution steps.
	<ol> <li>If the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</li> <li>Note: Use Cloud Native Core Network Function Data Collector tool for capturing the logs. For more information, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.</li> </ol>

# 6.3.1.14 OcnssfAlternateRouteServiceDown

Table 6-187 OcnssfAlternateRouteServiceDown

Field	Details
Description	'OCNSSF Alternate Route service is down'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : Ocnssf Alternate Route service down'
Severity	Critical
Condition	None of the pods of the Alternate Route service is available.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9032
Metric Used	'up'
	<b>Note:</b> This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.



Table 6-187 (Cont.) OcnssfAlternateRouteServiceDown

Field	Details
Recommended Actions	The alert is cleared when the Alternate Route service is available.  Note: The threshold is configurable in the alerts.yaml  Steps:
	Run the following command to check the orchestration logs of Alternate Route service and check for liveness or readiness probe failures:
	kubectl get po -n <namespace></namespace>
	Note the full name of the pod that is not running, and use it in the following command:
	kubectl describe pod <specific desired="" full="" name="" pod=""> -n <namespace></namespace></specific>
	Refer to the application logs on Kibana and filter based on Alternate Route service names. Check for ERROR WARNING logs related to thread exceptions.
	3. Depending on the failure reason, take the resolution steps.
	<ol> <li>If the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</li> <li>Note: Use Cloud Native Core Network Function Data Collector tool for capturing the logs. For more information, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.</li> </ol>

## 6.3.1.15 OcnssfAuditorServiceDown

Table 6-188 OcnssfAuditorServiceDown

Field	Describe.
Field	Details
Description	'OCNSSF NsAuditor service is down'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : Ocnssf NsAuditor service down'
Severity	Critical
Condition	None of the pods of the NsAuditor service is available.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9031
Metric Used	'up'
	<b>Note:</b> This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.



Table 6-188 (Cont.) OcnssfAuditorServiceDown

Field	Details
Recommended Actions	The alert is cleared when the NsAuditor service is available.
	Note: The threshold is configurable in the alerts.yaml
	Steps:
	Run the following command to check the orchestration logs of NsAuditor service and check for liveness or readiness probe failures:
	kubectl get po -n <namespace></namespace>
	Note the full name of the pod that is not running, and use it in the following command:
	kubectl describe pod <specific desired="" full="" name="" pod=""> -n <namespace></namespace></specific>
	Refer to the application logs on Kibana and filter based on NsAuditor service names. Check for ERROR WARNING logs related to thread exceptions.
	3. Depending on the failure reason, take the resolution steps.
	If the issue persists, capture all the outputs for the above steps and contact My Oracle Support.     Note: Use Cloud Native Core Network Function Data Collector tool for capturing the logs. For more information, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.

# 6.3.1.16 OcnssfTotalIngressTrafficRateAboveMinorThreshold

Table 6-189 OcnssfTotalIngressTrafficRateAboveMinorThreshold

Field	Details
Description	'Ingress traffic Rate is above the configured minor threshold i.e. 800 requests per second (current value is: {{ \$value }})'
Summary	'timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }}: Traffic Rate is above 80 Percent of Max requests per second(1000)'
Severity	Minor
Condition	The total Ocnssf Ingress Message rate has crossed the configured minor threshold of 800 TPS.
	Default value of this alert trigger point in NrfAlertValues.yaml is when Ocnssf Ingress Rate crosses 80 % of 1000 (Maximum ingress request rate).
OID	1.3.6.1.4.1.323.5.3.40.1.2.9010
Metric Used	'oc_ingressgateway_http_requests_total'



Table 6-189 (Cont.) OcnssfTotalIngressTrafficRateAboveMinorThreshold

Field	Details
Recommended Actions	The alert is cleared either when the total Ingress Traffic rate falls below the Minor threshold or when the total traffic rate crosses the Major threshold, in which case the OcnssfTotalIngressTrafficRateAboveMinorThreshold alert shall be raised.
	Note: The threshold is configurable in the alerts.yaml
	Steps:
	Reassess the reason why the NSSF is receiving additional traffic, for example, the mated site NSSF is unavailable in the georedundancy scenario.
	If this is unexpected, contact My Oracle Support.
	Refer Grafana to determine which service is receiving high traffic.
	<ol><li>Refer Ingress Gateway section in Grafana to determine an increase in 4xx and 5xx error codes.</li></ol>
	3. Check Ingress Gateway logs on Kibana to determine the reason for the errors.

# 6.3.1.17 OcnssfTotalIngressTrafficRateAboveMajorThreshold

Table 6-190 OcnssfTotalIngressTrafficRateAboveMajorThreshold

Field	Details
Description	'Ingress traffic Rate is above the configured major threshold i.e. 900 requests per second (current value is: {{ \$value }})'
Summary	'timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }}: Traffic Rate is above 90 Percent of Max requests per second(1000)'
Severity	Major
Condition	The total Ocnssf Ingress Message rate has crossed the configured major threshold of 900 TPS.
	Default value of this alert trigger point in NrfAlertValues.yaml is when Ocnssf Ingress Rate crosses 90 % of 1000 (Maximum ingress request rate).
OID	1.3.6.1.4.1.323.5.3.40.1.2.9011
Metric Used	'oc_ingressgateway_http_requests_total'
Recommended Actions	The alert is cleared when the total Ingress traffic rate falls below the major threshold or when the total traffic rate crosses the critical threshold, in which case the alert shall be raised.
	OcnssfTotalIngressTrafficRateAboveCriticalThreshold
	Note: The threshold is configurable in the alerts.yaml
	Steps:
	Reassess the reason why the NSSF is receiving additional traffic, for example, the mated site NSSF is unavailable in the georedundancy scenario.
	If this is unexpected, contact My Oracle Support.
	Refer Grafana to determine which service is receiving high traffic.
	Refer Ingress Gateway section in Grafana to determine an increase in 4xx and 5xx error codes.
	3. Check Ingress Gateway logs on Kibana to determine the reason for the errors.



## $6.3.1.18\ Ocnss f Total Ingress Traffic Rate Above Critical Threshold$

Table 6-191 OcnssfTotalIngressTrafficRateAboveCriticalThreshold

Field	Details
Description	'Ingress traffic Rate is above the configured critical threshold i.e. 950 requests per second (current value is: {{ \$value }})'
Summary	'timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }}: Traffic Rate is above 95 Percent of Max requests per second(1000)'
Severity	Critical
Condition	The total Ocnssf Ingress Message rate has crossed the configured critical threshold of 950 TPS.
	Default value of this alert trigger point in NrfAlertValues.yaml is when Ocnssf Ingress Rate crosses 95 % of 1000 (Maximum ingress request rate).
OID	1.3.6.1.4.1.323.5.3.40.1.2.9012
Metric Used	'oc_ingressgateway_http_requests_total'
Recommended Actions	The alert is cleared when the Ingress traffic rate falls below the critical threshold.
	Note: The threshold is configurable in the alerts.yaml
	Steps:
	Reassess the reason why the NSSF is receiving additional traffic, for example, the mated site NSSF is unavailable in the georedundancy scenario.
	If this is unexpected, contact My Oracle Support.
	Refer Grafana to determine which service is receiving high traffic.
	<ol> <li>Refer Ingress Gateway section in Grafana to determine an increase in 4xx and 5xx error codes.</li> </ol>
	3. Check Ingress Gateway logs on Kibana to determine the reason for the errors.

### 6.3.1.19 OcnssfTransactionErrorRateAbove0.1Percent

Table 6-192 OcnssfTransactionErrorRateAbove0

Field	Details
Description	'Transaction Error rate is above 0.1 Percent of Total Transactions (current value is {{ \$value }})'
Summary	'timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }}: Transaction Error Rate detected above 0.1 Percent of Total Transactions'
Severity	Warning
Condition	The number of failed transactions is above 0.1 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9013
Metric Used	'oc_ingressgateway_http_responses_total'



Table 6-192 (Cont.) OcnssfTransactionErrorRateAbove0

Field	Details
Recommended Actions	The alert is cleared when the number of failure transactions are below 0.1 percent of the total transactions or when the number of failure transactions crosses the 1% threshold in which case the OcnssfTransactionErrorRateAbove1Percent is raised.
	Steps:
	Check the Service specific metrics to understand the specific service request errors.  For example: ocnssf_nsselection_success_tx_total with statusCode ~= 2xx.
	Verify the metrics per service, per method.     For example: Discovery requests can be deduced from the follwing metrics:
	Metrics="oc_ingressgateway_http_responses_total"  Method="GET"
	NFServiceType="ocnssf-nsselection"
	Route_path="/nnssf-nsselection/v2/**"
	Status="503 SERVICE_UNAVAILABLE"
	3. If guidance is required, contact My Oracle Support.

### 6.3.1.20 OcnssfTransactionErrorRateAbove1Percent

Table 6-193 OcnssfTransactionErrorRateAbove1Percent

Field	Details
Description	'Transaction Error rate is above 1 Percent of Total Transactions (current value is {{ \$value }})'
Summary	'timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }}: Transaction Error Rate detected above 1 Percent of Total Transactions'
Severity	Warning
Condition	The number of failed transactions is above 1 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9014
Metric Used	'oc_ingressgateway_http_responses_total'
Recommended Actions	The alert is cleared when the number failed transactions is below 1% of the total transactions or when the number of failed transactions crosses the 10% threshold in which case the OcnssfTransactionErrorRateAbove10Percent shall be raised.  Steps:
	<ol> <li>Check the Service specific metrics to understand the specific service request errors.         For example: ocnssf_nsselection_success_tx_total with statusCode ~= 2xx.</li> <li>Verify the metrics per service, per method         For example: Discovery requests can be deduced from the following metrics:         Metrics="oc_ingressgateway_http_responses_total"         Method="GET"         NFServiceType="ocnssf-nsselection"</li> </ol>
	Route_path="/nnssf-nsselection/v2/**" Status="503 SERVICE_UNAVAILABLE"  3. If guidance is required, contact My Oracle Support.



### 6.3.1.21 OcnssfTransactionErrorRateAbove10Percent

Table 6-194 OcnssfTransactionErrorRateAbove10Percent

Field	Details
Description	'Transaction Error rate is above 10 Percent of Total Transactions (current value is {{ \$value }})'
Summary	'timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }}: Transaction Error Rate detected above 10 Percent of Total Transactions'
Severity	Minor
Condition	The number of failed transactions has crossed the minor threshold of 10 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9015
Metric Used	'oc_ingressgateway_http_responses_total'
Recommended Actions	The alert is cleared when the number of failed transactions crosses the 10% threshold of the total transactions or when the ailed transactions crosses the 25% threshold in which case the OcnssfTransactionErrorRateAbove25Percent shall be raised.
	Steps:
	Check the Service specific metrics to understand the specific service request errors.  For example: ocnssf_nsselection_success_tx_total with statusCode ~= 2xx.
	2. Verify the metrics per service, per method For example: Discovery requests can be deduced from the following metrics: Metrics="oc_ingressgateway_http_responses_total" Method="GET" NFServiceType="ocnssf-nsselection" Route_path="/nnssf-nsselection/v2/**" Status="503 SERVICE_UNAVAILABLE"  3. If guidance is required, contact My Oracle Support.

### 6.3.1.22 OcnssfTransactionErrorRateAbove25Percent

Table 6-195 OcnssfTransactionErrorRateAbove25Percent

Field	Details
1 1014	Beans
Description	'Transaction Error rate is above 25 Percent of Total Transactions (current value is {{ \$value }})'
summary	'timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }}: Transaction Error Rate detected above 25 Percent of Total Transactions'
Severity	Major
Condition	The number of failed transactions has crossed the minor threshold of 25 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9016
Metric Used	'oc_ingressgateway_http_responses_total'



Table 6-195 (Cont.) OcnssfTransactionErrorRateAbove25Percent

Field	Details
Recommended Actions	The alert is cleared when the number of failed transactions crosses the 25% of the total transactions or when the number of failed transactions crosses the 50% threshold in which case the OcnssfTransactionErrorRateAbove50Percent shall be raised.
	Steps:
	Check the Service specific metrics to understand the specific service request errors.  For example: ocnssf_nsselection_success_tx_total with statusCode ~= 2xx.
	Verify the metrics per service, per method     For example: Discovery requests can be deduced from the following metrics:
	Metrics="oc_ingressgateway_http_responses_total"
	Method="GET"
	NFServiceType="ocnssf-nsselection"
	Route_path="/nnssf-nsselection/v2/**"
	Status="503 SERVICE_UNAVAILABLE"
	3. If guidance is required, contact My Oracle Support.

### 6.3.1.23 OcnssfTransactionErrorRateAbove50Percent

Table 6-196 OcnssfTransactionErrorRateAbove50Percent

Field	Details
Description	'Transaction Error rate is above 50 Percent of Total Transactions (current value is {{ \$value }})'
Summary	'timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }}: Transaction Error Rate detected above 50 Percent of Total Transactions'
Severity	Critical
Condition	The number of failed transactions has crossed the minor threshold of 50 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9017
Metric Used	'oc_ingressgateway_http_responses_total



Table 6-196 (Cont.) OcnssfTransactionErrorRateAbove50Percent

Field	Details
Recommended Actions	The alert is cleared when the number of failed transactions is below 50 percent of the total transactions.
	Steps:
	Check for service specific metrics to understand the specific service request errors.  For example: ocnssf_nsselection_success_tx_total with statusCode ~= 2xx.
	Verify the metrics per service, per method     For example: Discovery requests can be deduced from the following metrics:
	Metrics="oc_ingressgateway_http_responses_total"  Method="GET"
	NFServiceType="ocnssf-nsselection"
	Route_path="/nnssf-nsselection/v2/**"
	Status="503 SERVICE_UNAVAILABLE"
	3. If guidance is required, contact My Oracle Support.

# 6.3.1.24 OcnssfIngressGatewayPodCongestionStateWarning

Table 6-197 OcnssfIngressGatewayPodCongestionStateWarning

Field	Details
Description	Ingress gateway pod congestion state reached DOC
Summary	'timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }}: Ingress gateway pod congestion state reached DOC'
Severity	Warning
Condition	Ingress gateway pod has moved into a state of DOC for any of the aforementioned metrics. Thresholds are configured for CPU, Pending messages count.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9027
Metric Used	oc_ingressgateway_pod_congestion_state
Recommended Actions	Reassess the reasons leading to NSSF receiving additional traffic.
	Steps:
	Reassess the reasons leading to NSSF receiving additional traffic.
	If this is unexpected, contact My Oracle Support.
	1. Refer to alert to determine which service is receiving high traffic. It may be due to a sudden spike in traffic.
	For example: When one mated site goes down, the NFs move to the given site.
	2. Check the service pod logs on Kibana to determine the reason for the errors.
	3. If this is expected traffic, check in Grafana for the distribution of traffic among the Ingress gateway pods. Then the thresholds levels may be reevaluated as per the call rate and reconfigured as mentioned in <i>Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide</i> .



## 6.3.1.25 OcnssfIngressGatewayPodCongestionStateMajor

Table 6-198 OcnssflngressGatewayPodCongestionStateMajor

Field	Details
Description	Ingress gateway pod congestion state when reached CONGESTED
Summary	'timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }}: Ingress gateway pod congestion state when reached CONGESTED'
Severity	Major
Condition	Ingress gateway pod has moved into a state of CONGESTED for any of the aforementioned metrics. Thresholds are configured for CPU, Pending messages count.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9028
Metric Used	oc_ingressgateway_pod_congestion_state
Recommended Actions	Reassess the reasons leading to NSSF receiving additional traffic.
	Steps:
	Reassess the reasons leading to NSSF receiving additional traffic.
	If this is unexpected, contact My Oracle Support.
	1. Refer to alert to determine which service is receiving high traffic. It may be due to a sudden spike in traffic.
	For example: When one mated site goes down, the NFs move to the given site.
	2. Check the service pod logs on Kibana to determine the reason for the errors.
	3. If this is expected traffic, check in Grafana for the distribution of traffic among the Ingress gateway pods. Then the thresholds levels may be reevaluated as per the call rate and reconfigured as mentioned in <i>Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.</i>

## 6.3.1.26 OcnssfIngressGatewayPodResourceStateWarning

Table 6-199 OcnssflngressGatewayPodResourceStateWarning

Field	Details
Description	The ingress gateway pod congestion state reached DOC because of excessive usage of resources
Summary	'timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }}: The ingress gateway pod congestion state reached DOC because of excessive usage of resources'
Severity	Warning
Condition	The configured threshold for resource cunsumption for state DOC for Ingress gateway is breached.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9029
Metric Used	oc_ingressgateway_pod_resource_state



Table 6-199 (Cont.) OcnssflngressGatewayPodResourceStateWarning

Field	Details
Recommended Actions	Reassess the reasons leading to NSSF receiving additional traffic.
	Steps:
	Reassess the reasons leading to NSSF receiving additional traffic.
	If this is unexpected, contact My Oracle Support.
	1. Refer to alert to determine which service is receiving high traffic. It may be due to a sudden spike in traffic.
	For example: When one mated site goes down, the NFs move to the given site.
	2. Check the service pod logs on Kibana to determine the reason for the errors.
	3. If this is expected traffic, check in Grafana for the distribution of traffic among the Ingress gateway pods. Then the thresholds levels may be reevaluated as per the call rate and reconfigured as mentioned in <i>Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.</i>

## 6.3.1.27 OcnssfIngressGatewayPodResourceStateMajor

Table 6-200 OcnssflngressGatewayPodResourceStateMajor

Field	Details
Description	The ingress gateway pod congestion state reached CONGESTED because of excessive usage of resources
Summary	'timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }}: The ingress gateway pod congestion state reached CONGESTED because of excessive usage of resources'
Severity	Major
Condition	The configured threshold for resource cunsumption for state CONGESTED for Ingress gateway is breached.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9030
Metric Used	oc_ingressgateway_pod_resource_state
Recommended Actions	Reassess the reasons leading to NSSF receiving additional traffic.
	Steps:
	Reassess the reasons leading to NSSF receiving additional traffic.
	If this is unexpected, contact My Oracle Support.
	1. Refer to alert to determine which service is receiving high traffic. It may be due to a sudden spike in traffic.
	For example: When one mated site goes down, the NFs move to the given site.
	2. Check the service pod logs on Kibana to determine the reason for the errors.
	3. If this is expected traffic, check in Grafana for the distribution of traffic among the Ingress gateway pods. Then the thresholds levels may be reevaluated as per the call rate and reconfigured as mentioned in <i>Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide</i> .

# 6.3.2 Application Level Alerts

This section lists the application level alerts.



## 6.3.2.1 ocnssfPolicyNotFoundWarning

Table 6-201 ocnssfPolicyNotFoundWarning

Field	Details
Description	'Policy Not Found Rate is above warning threshold i.e. 700 mps (current value is: {{ \$value }})'
Summary	'timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }}: 'Policy Not Found Rate is above 70 Percent' rate(ocnssf_nsselection_policy_not_found_total[2m])) >= 100 < 150
Severity	Warning
Condition	Rate of messages that did not find a matching policy is above warning threshold (Threshold: <>, Current: <>).
OID	1.3.6.1.4.1.323.5.3.40.1.2.9018
Metric Used	ocnssf_nsselection_policy_not_found_total
Recommended Actions	This alert is cleared when the number of error transactions are below 70 percent of the total traffic.
	Steps:
	Check the ocnssf_nsselection_policy_match_total rate.
	2. Look into logs and find configuration mismatch:
	a. For failure scenario check TAI and SNSSAI in error logs.
	b. Look in configuration for corresponding nssai-auth and nss_rule.
	i. If entry is not found, add configuration.
	ii. If entry is found, check Grant_FileId and update to ALLOWED.
	3. If guidance is required, contact My Oracle Support.

# $6.3.2.2\ ocnssfPolicyNotFoundMajor$

Table 6-202 ocnssfPolicyNotFoundMajor

Field	Details
Description	'Policy Not Found Rate is above major threshold i.e. 850 mps (current value is: {{ \$value }})'
Summary	'timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }}: 'Policy Not Found Rate is above 85 Percent'
Severity	Major
Condition	Rate of messages that did not find a matching policy is above major threshold.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9019
Metric Used	ocnssf_nsselection_policy_not_found_total



Table 6-202 (Cont.) ocnssfPolicyNotFoundMajor

Field	Details
Recommended Actions	This alert is cleared when the number of error transactions are below 85% of the total traffic.
	Steps:
	Check the ocnssf_nsselection_policy_match_total rate.
	2. Look into logs and find configuration mismatch:
	a. For failure scenario check TAI and SNSSAI in error logs.
	b. Look in configuration for corresponding nssai-auth and nss_rule.
	i. If entry is not found, add configuration.
	ii. If entry is found, check Grant_Fileld and update to ALLOWED.
	3. If guidance is required, contact My Oracle Support.

# 6.3.2.3 ocnssfPolicyNotFoundCritical

Table 6-203 ocnssfPolicyNotFoundCritical

Field	Description
Description	'Policy Not Found Rate is above critical threshold i.e. 950 mps (current value is: {{ \$value }})'
Summary	'timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }}: 'Policy Not Found Rate is above 95 Percent'
Severity	Critical
Condition	Rate of messages that did not find a matching policy is above critical threshold.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9020
Metric Used	ocnssf_nsselection_policy_not_found_total
Recommended Actions	This alert is cleared when the number of error transactions are below 95 percent of the total traffic.
	Steps:
	Check the ocnssf_nsselection_policy_match_total rate
	2. Look into logs and find configuration mismatch:
	<ul> <li>a. For failure scenario check TAI and SNSSAI in error logs.</li> </ul>
	b. Look in configuration for corresponding nssai-auth and nss_rule:
	i. If entry is not found, add configuration.
	ii. If entry is found, check Grant_FileId and update to ALLOWED.
	3. If guidance is required, contact My Oracle Support.



### 6.3.2.4 OcnssfOverloadThresholdBreachedL1

Table 6-204 OcnssfOverloadThresholdBreachedL1

Field	Details
Description	'Overload Level of {{\$labels.app_kubernetes_io_name}} service is L1'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, podname: {{\$labels.kubernetes_pod_name}}: Overload Level of {{\$labels.app_kubernetes_io_name}} service is L1'
Severity	Warning
Condition	NSSF Services have breached their configured threshold of Level L1 for any of the aforementioned metrics.  Thresholds are configured for CPU, svc_failure_count, svc_pending_count, and memory.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9021
Metric Used	load_level
Recommended Actions	The alert is cleared when the Ingress Traffic rate falls below the configured L1 threshold.
	Note: The thresholds can be configured using REST API.
	Steps:
	Reassess the reasons leading to NSSF receiving additional traffic.
	If this is unexpected, contact My Oracle Support.
	1. Refer to alert to determine which service is receiving high traffic. It may be due to a sudden spike in traffic.
	For example: When one mated site goes down, the NFs move to the given site.
	2. Check the service pod logs on Kibana to determine the reason for the errors.
	3. If this is expected traffic, then the thresholds levels may be reevaluated as per the call rate and reconfigured as mentioned in <i>Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.</i>

### 6.3.2.5 OcnssfOverloadThresholdBreachedL2

Table 6-205 OcnssfOverloadThresholdBreachedL2

Field	Details
Description	'Overload Level of {{\$labels.app_kubernetes_io_name}} service is L2'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, podname: {{\$labels.kubernetes_pod_name}}: Overload Level of {{\$labels.app_kubernetes_io_name}} service is L2'
Severity	Warning
Condition	NSSF Services have breached their configured threshold of Level L2 for any of the aforementioned metrics.  Thresholds are configured for CPU, svc_failure_count, svc_pending_count, and memory.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9022
Metric Used	load_level



Table 6-205 (Cont.) OcnssfOverloadThresholdBreachedL2

Field	Details
Recommended Actions	The alert is cleared when the Ingress Traffic rate falls below the configured L2 threshold.
	Note: The thresholds can be configured using REST API.
	Steps:
	Reassess the reasons leading to NSSF receiving additional traffic.
	If this is unexpected, contact My Oracle Support.
	1. Refer to alert to determine which service is receiving high traffic. It may be due to a sudden spike in traffic.
	For example: When one mated site goes down, the NFs move to the given site.
	2. Check the service pod logs on Kibana to determine the reason for the errors.
	3. If this is expected traffic, then the thresholds levels may be reevaluated as per the call rate and reconfigured as mentioned in <i>Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.</i>

### 6.3.2.6 OcnssfOverloadThresholdBreachedL3

Table 6-206 OcnssfOverloadThresholdBreachedL3

Field	Details
Description	'Overload Level of {{\$labels.app_kubernetes_io_name}} service is L3'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, podname: {{\$labels.kubernetes_pod_name}}: Overload Level of {{\$labels.app_kubernetes_io_name}} service is L3'
Severity	Warning
Condition	NSSF Services have breached their configured threshold of Level L3 for any of the aforementioned metrics.  Thresholds are configured for CPU, svc_failure_count, svc_pending_count, and memory.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9023
Metric Used	load_level
Recommended Actions	The alert is cleared when the Ingress Traffic rate falls below the configured L3 threshold.
	Note: The thresholds can be configured using REST API.
	Steps:
	Reassess the reasons leading to NSSF receiving additional traffic.
	If this is unexpected, contact My Oracle Support.
	1. Refer to alert to determine which service is receiving high traffic. It may be due to a sudden spike in traffic.
	For example: When one mated site goes down, the NFs move to the given site.
	2. Check the service pod logs on Kibana to determine the reason for the errors.
	3. If this is expected traffic, then the thresholds levels may be reevaluated as per the call rate and reconfigured as mentioned in <i>Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.</i>



### 6.3.2.7 OcnssfOverloadThresholdBreachedL4

Table 6-207 OcnssfOverloadThresholdBreachedL4

Field	Details
Description	'Overload Level of {{\$labels.app_kubernetes_io_name}} service is L4'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, podname: {{\$labels.kubernetes_pod_name}}: Overload Level of {{\$labels.app_kubernetes_io_name}} service is L4'
Severity	Warning
Condition	NSSF Services have breached their configured threshold of Level L4 for any of the aforementioned metrics.  Thresholds are configured for CPU, svc_failure_count, svc_pending_count, and memory.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9024
Metric Used	load_level
Recommended Actions	The alert is cleared when the Ingress Traffic rate falls below the configured L4 threshold.
	Note: The thresholds can be configured using REST API.
	Steps:
	Reassess the reasons leading to NSSF receiving additional traffic.
	If this is unexpected, contact My Oracle Support.
	1. Refer to alert to determine which service is receiving high traffic. It may be due to a sudden spike in traffic.
	For example: When one mated site goes down, the NFs move to the given site.
	2. Check the service pod logs on Kibana to determine the reason for the errors.
	3. If this is expected traffic, then the thresholds levels may be reevaluated as per the call rate and reconfigured as mentioned in <i>Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.</i>

# 6.3.2.8 OcnssfScpMarkedAsUnavailable

Table 6-208 OcnssfScpMarkedAsUnavailable

Field	Details
Description	'An SCP has been marked unavailable'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : One of the SCP has been marked unavailable'
Severity	Major
Condition	One of the SCPs has been marked unhealthy.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9025
Metric Used	'oc_egressgateway_peer_health_status'
Recommended Actions	This alert get cleared when unavailable SCPs become available.



### 6.3.2.9 OcnssfAllScpMarkedAsUnavailable

Table 6-209 OcnssfAllScpMarkedAsUnavailable

Field	Details
Description	'All SCPs have been marked unavailable'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : All SCPs have been marked as unavailable'
Severity	Critical
Condition	All SCPs have been marked unavailable.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9026
Metric Used	'oc_egressgateway_peer_count and oc_egressgateway_peer_available_count'
Recommended Actions	NF clears the critical alarm when at least one SCP peer in a peer set becomes available such that all other SCP or SEPP peers in the given peer set are still unavailable.

## 6.3.2.10 OcnssfTLSCertificateExpireMinor

Table 6-210 OcnssfTLSCertificateExpireMinor

Field	Details
Description	'TLS certificate to expire in 6 months'.
Summary	'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : TLS certificate to expire in 6 months'
Severity	Minor
Condition	This alert is raised when the TLS certificate is about to expire in six months.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9038
Metric Used	security_cert_x509_expiration_seconds
Recommended Actions	The alert is cleared when the TLS certificate is renewed.
	For more information about certificate renewal, see "Creating Private Keys and Certificate " section in the Oracle Communications Cloud Native Core, Network Slice Selection Function Installation, Upgrade, and Fault Recovery Guide.

### 6.3.2.11 OcnssfTLSCertificateExpireMajor

Table 6-211 OcnssfTLSCertificateExpireMajor

Field	Details
Description	'TLS certificate to expire in 3 months.'
Summary	'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : TLS certificate to expire in 3 months'
Severity	Major
Condition	This alert is raised when the TLS certificate is about to expire in three months.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9039
Metric Used	security_cert_x509_expiration_seconds



#### Table 6-211 (Cont.) OcnssfTLSCertificateExpireMajor

Field	Details
Recommended Actions	The alert is cleared when the TLS certificate is renewed.
	For more information about certificate renewal, see "Creating Private Keys and Certificate " section in the Oracle Communications Cloud Native Core, Network Slice Selection Function Installation, Upgrade, and Fault Recovery Guide.

# $6.3.2.12\ OcnssfTLSCertificate Expire Critical$

Table 6-212 OcnssfTLSCertificateExpireCritical

Field	Details
Description	'TLS certificate to expire in one month.'
Summary	'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : TLS certificate to expire in 1 month'
Severity	Critical
Condition	This alert is raised when the TLS certificate is about to expire in one month.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9040
Metric Used	security_cert_x509_expiration_seconds
Recommended Actions	The alert is cleared when the TLS certificate is renewed.
	For more information about certificate renewal, see "Creating Private Keys and Certificate" section in the Oracle Communications Cloud Native Core, Network Slice Selection Function Installation, Upgrade, and Fault Recovery Guide.

## 6.3.2.13 OcnssfNrfInstancesInDownStateMajor

Table 6-213 OcnssfNrfInstancesInDownStateMajor

Field	Details
Description	'When current operative status of any NRF Instance is unavailable/unhealthy'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : Few of the NRF instances are in unavailable state'
Severity	Major
Condition	When sum of the metric values of each NRF instance is greater than 0 but less than 3.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9042
Metric Used	nrfclient_nrf_operative_status



Table 6-213 (Cont.) OcnssfNrfInstancesInDownStateMajor

Field	Details
Recommended Actions	This alert is cleared when operative status of all the NRF Instances is available/ healthy.
	Steps:
	Check the nrfclient_nrf_operative_status metric value of each NRF instance.
	2. The instances for which the metric value is '0' are down.
	3. Bring up the NRF instances that are down.
	<ol> <li>If the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</li> </ol>
	<b>Note:</b> Use Cloud Native Core Network Function Data Collector tool for capturing the logs. For more information, see <i>Oracle Communications Cloud Native Core</i> , <i>Network Function Data Collector User Guide</i> .

### 6.3.2.14 OcnssfAllNrfInstancesInDownStateCritical

Table 6-214 OcnssfAllNrfInstancesInDownStateCritical

Field	Details
Description	'When current operative status of all the NRF Instances is unavailable/unhealthy'
Summary	'kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ .   first   value   humanizeTimestamp }}{{ end }} : All the NRF instances are in unavailable state'
Severity	Critical
Condition	When sum of the metric values of each NRF instance is equal to 0.
OID	1.3.6.1.4.1.323.5.3.40.1.2.9041
Metric Used	nrfclient_nrf_operative_status
Recommended Actions	This alert is cleared when current operative status of atleast one NRF Instance is available/healthy.
	Steps:
	Bring up at least one NRF Instance.
	<ol> <li>If the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</li> <li>Note: Use Cloud Native Core Network Function Data Collector tool for capturing the logs. For more information, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.</li> </ol>

# 6.3.3 NSSF Alert Configuration

Follow the steps below for NSSF Alert configuration in Prometheus:



#### (i) Note

- 1. By default, Namespace for NSSF is ocnssf, which must be updated in the ocnssf\_alert\_rules\_25.1.100.yaml or ocnssf\_alert\_rules\_promha\_25.1.100.yaml files as per the deployment.
- 2. The ocnssf-custom-configtemplates-25\_1\_100\_0\_0 file can be downloaded from MOS. Unzip the ocnssf-custom-configtemplates-25\_1\_100\_0\_0 file after downloading to get ocnssf\_custom\_values\_25.1.100.yaml file.
- 3. Set the following parameter in the ocnssf\_alert\_rules\_25.1.100.yaml file: app\_kubernetes\_io\_part\_of="<deployment name>"

Example: app\_kubernetes\_io\_part\_of="ocnssf"

Where deployment name is 'ocnssf'.

#### Configuring NSSF alerts for CNE 1.8.x and previous versions

The following procedure describes how to configure NSSF alerts for CNE version 1.8.x and previous versions:

**NAME**: Helm Release of Prometheus

**\_Namespace\_**: Kubernetes Namespace in which Prometheus is installed

1. Take a backup of current configuration map of Prometheus:

```
kubectl get configmaps _NAME_-server -o yaml -n _Namespace_ > /tmp/
tempConfig.yaml
```

2. Check NSSF Alert file name:

```
sed -i '/etc\/config\/alertsnssf/d' /tmp/tempConfig.yaml
```

3. Add NSSF Alert file name inside Prometheus configuration map:

```
sed -i '/rule_files:/a\ \- /etc/config/alertsnssf'/tmp/tempConfig.yaml
```

4. Update configuration map with the updated file name of NSSF alert file:

```
kubectl replace configmap _NAME_-server -f /tmp/tempConfig.yaml
```

5. Run the following command to change the default namespace to a new namespace.

#### (i) Note

If you do not change the default namespace, it will show the default namespace under alerts on Prometheus dashboard.

sed -i "s/<default-namespace>/<new-namespace>/g"
ocnssf\_alert\_rules\_<release-version>.yaml



#### Example:

sed -i "s/ocnssf-namespace/ocnssf/g" ocnssf\_alert\_rules\_25.1.100.yaml

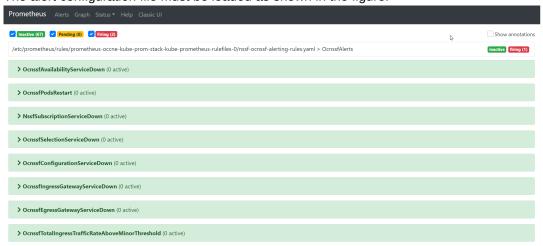
6. Add NSSF Alert rules in configuration map under file name of NSSF alert file:

```
kubectl patch configmap _NAME_-server -n _Namespace_--type merge --patch "$
(cat ~/ocnssf alert rules <release-version>.yaml)"
```

#### Example:

```
kubectl patch configmap _NAME_-server -n _Namespace_--type merge --patch "$
(cat ~/ocnssf_alert_rules_25.1.100.yaml)"
```

Log in to Prometheus GUI and verify the alerts section.The alert configuration file must be loaded as shown in the figure.



#### Configuring NSSF alerts for CNE 1.9.x and later versions

This section describes the measurement based Alert rules configuration for NSSF in Prometheus. Use the ocnssf\_alerting\_rules\_promha\_<release-number>.yaml file updated in NSSF Alert configuration section.

1. Run the following command to change the default namespace to a new namespace.



If you do not change the default namespace, it will show the default namespace under alerts on Prometheus dashboard.

```
sed -i "s/<default-namespace>/<new-namespace>/g"
ocnssf_alert_rules_promha_<release-version>.yaml
```

#### For example:

sed -i "s/ocnssf-namespace/ocnssf/g"
ocnssf\_alert\_rules\_promha\_25.1.100.yaml



2. Run the following command to apply the prometheus rules:

kubectl apply -f ocnssf\_alert\_rules\_promha\_<release-version>.yaml -n
\_Namespace\_

#### Example:

\$ kubectl apply -f ocnssf\_alert\_rules\_promha\_25.1.100.yaml --namespace
ocnssf

#### Sample output:

prometheusrule.monitoring.coreos.com/ocnssf-alerting-rules created

- 3. Run the following command to check NSSF alert file is added to prometheusrules:
  - \$ kubectl get prometheusrules --namespace <namespace>

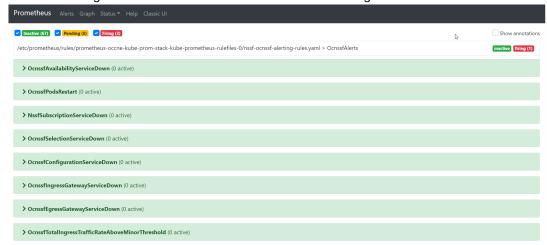
#### Example:

\$ kubectl get prometheusrules --namespace ocnssf

#### Sample output:

NAME AGE nssf-ocnssf-alerting-rules 1m

Log in to Prometheus GUI and verify the alerts section.
The alert configuration file must be loaded as shown in the figure.





#### ① Note

The Prometheus server takes an updated configuration map that is automatically reloaded after approximately 60 seconds. Refresh the Prometheus GUI to confirm that the NSSF Alerts have been reloaded.

#### Steps to Check Alerts in Prometheus

1. Run the following command to deploy Prometheus:

Go to path: ocnssf/automation/infrastructure

```
helm install stable/Prometheus occne-prometheus --namespace occne-infra -f ./components/prometheus/values.yaml -f./components/prometheus/values-server-files.yaml --version 9.1.1
```

- 2. Configure the alerts by following NSSF Alert Configuration section.
- 3. To find Prometheus on UI:

```
http://_NODE_IP_:PORT_/
```

Here \_NODE\_IP\_ is the machine on which Prometheus pod is running.

PORT is occne-prometheus-server port. (Use cmd :: "kubectl get svc -n occne-infra" to get port. Here occne-infra is the namespace where Prometheus is running.)

#### **Disabling Alerts**

This section explains the procedure to disable the alerts in NSSF.

#### Disabling Alerts for CNE 1.8.x and previous versions

- 1. Edit ocnssf\_alert\_rules\_25.1.100.yaml file to remove specific alert.
- 2. Remove complete content of the specific alert from the ocnssf alert rules 25.1.100.yaml file.

```
ocnssf_alert_rules_25.1.100.yaml
```

```
alert, remove the complete content:

## ALERT SAMPLE START##
- alert: OcnssfTrafficRateAboveMinorThreshold
annotations:
description: 'NSSF traffic Rate is above the configured minor threshold
i.e. 700 requests per second (current value is: {{ $value }})'
summary: 'namespace: {{$labels.kubernetes_namespace}}, podname:
{{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ .
  | first | value | humanizeTimestamp }}{{ .
  | end }}: Traffic Rate is above 70
Percent of Max requests per second(1000)'
expr:
sum(rate(oc_ingressgateway_http_requests_total{InstanceIdentifier="nssf_ingressgateway",kubernetes_namespace="nssf"}[2m])) > 0
labels:
severity: minor
```

For example: If you want to remove OcnssfTrafficRateAboveMinorThreshold



```
oid: "1.3.6.1.4.1.323.5.3.51.1.2.7001"
namespace: ' {{ $labels.kubernetes_namespace }} '
podname: ' {{$labels.kubernetes_pod_name}} '
## ALERT SAMPLE END##
```

3. Perform Alert configuration.

#### Disabling Alerts for CNE 1.9.x and later versions

1. Retrieve prometheusrule name.

#### Example:

Run

kubectl get prometheusrule

#### Sample Output:

```
NAME AGE ocnssf-alerting-rules 7d20h
```

Delete prometheusrule.

#### Example:

Run

kubectl delete prometheusrule ocnssf-alerting-rules

#### Sample Output:

prometheusrule.monitoring.coreos.com "ocnssf-alerting-rules" deleted

3. Update alerting rules.

Example:

Run

```
kubectl apply -f promHAalerts.yaml
```

#### Sample Output:

prometheusrule.monitoring.coreos.com/ocnssf-alerting-rules created

## 6.3.4 Configuring SNMP Notifier

This section describes the procedure to configure SNMP Notifier.

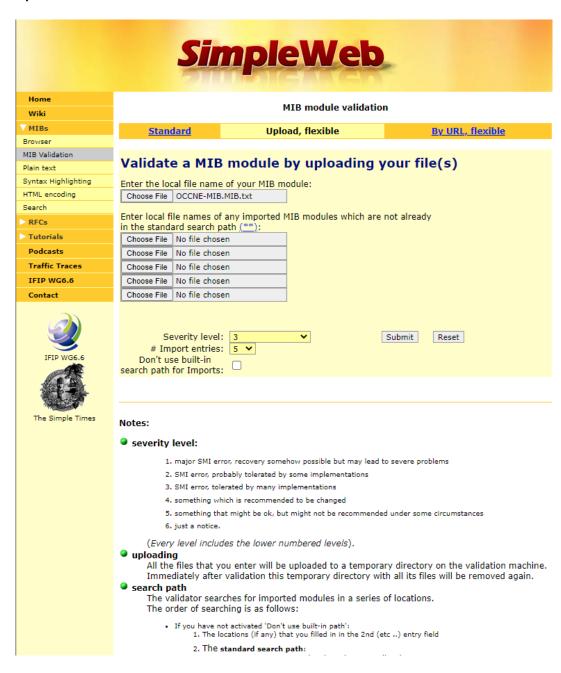
The SNMP MIB files are used to define the MIB objects. When uploaded to Wireshark and MIB tools, such as MIB Browser and Trap Receiver, users can see the detailed MIB definition instead of just the OID. All tools require the valid syntax of MIB files, and even minor errors can cause the upload to fail.

#### **Procedure to Validate MIB Files**

This procedure explains how to validate the MIB files and how to fix some common errors.



- Download MIB Files: Download the MIB files onto your PC. In the NSSF environment, the files are named as follows:
  - NSSF-MIB.mib
  - NSSF-TC.mib
  - TEKELEC-TOPLEVEL-REG.mib
     These files are located in the path /ocnssf/observability/mib.
- 2. Open Simpleweb MIB Validator: Open the Simpleweb MIB validator page.
- 3. Upload MIB Files:



a. Under "Enter the local file name of your MIB module," click the "Choose File" button.



b. Select the MIB file you want to validate, click "Open," and the file will be added to the web page.

#### 4. Inspect MIB Definitions:

- a. Open the MIB file on your PC using any suitable application.
- b. In the "IMPORTS" section, identify the MIB definitions listed after the word "FROM."
- c. Skip standard MIBs such as "SNMPv2-SMI," "SNMPv2-TC," etc., as they are already included in the Simpleweb MIB validator by default.

#### 5. Handling Private MIBs:

- For other MIBs, especially private MIB files, locate the corresponding MIB file for each definition.
- b. If the MIB file names differ from the MIB DEFINITIONS, rename the file as <MIB definition>.mib. For example:
  - Original MIB file name: Private-MIB-File.mib
  - MIB definition in "IMPORTS" section: FROM PRIVATE-MIB Rename the file to: PRIVATE-MIB.mib

#### 6. Upload Corrected File Names:





- Upload the corrected file names into the Simpleweb MIB validator. In NSSF, the corrected file names will be NSSF\_MIB.mib, NSSF-TC.mib, TEKELEC-TOPLEVEL-REG.mib.
- b. Click "Choose Files" and then click "Submit" to complete the process.

By following these steps, you can ensure the proper validation of MIB files, including the handling of standard and private MIBs.

#### **Fixes for Common MIB Compliance Issues**

#### Import SNMPv2-SMI:

Message: "Invalid status 'current' in SMIv1 MIB"

Fix Method: Add "MODULE-IDENTITY FROM SNMPv2-SMI"

Example format:

#### **IMPORTS**

```
TEXTUAL-CONVENTION FROM SNMPv2-TC MODULE-IDENTITY FROM SNMPv2-SMI oracleCNE FROM TEKELEC-TOPLEVEL-REG;
```

#### Last Update and Revision:

#### Messages:

- "Revision date after last update"
- "Revision not in reverse chronological order"
- "Revision for the last update is missing"

#### Fix Method:

- Ensure LAST-UPDATED is exactly the same as the most recent REVISION.
- Create a separate "REVISION HISTORY" section.
- Put all REVISIONs in this section in reverse chronological order.

#### Example format:

```
oracleNssfMIB MODULE-IDENTITY
   LAST-UPDATED "202302091734Z"
   ...
   REVISION "202302091734Z"
   DESCRIPTION "Updated."
   ::= { oracleNSSF 1 }
```

#### **Case-Sensitive Names:**

Message: "<name> should start with a lowercase letter"

Fix Method: Change the first letter to lowercase.



#### Example format:

```
OCNSSFConfigurationServiceDown NOTIFICATION-TYPE ...

Fix:

ocnssfConfigurationServiceDown NOTIFICATION-TYPE ...
```

#### **Duplicate OID:**

Message: "Identifier ocnssfIngressGatewayServiceDown' registers object identifier already registered by ocnssfConfigurationServiceDown' "

Fix Method: Change the OID to an unused number.

Example format:

::= { oracleNssfMIBNotifications 9036 }

END