Cisco Optical Network Controller Infra APIs

CONTENTS

Cisco Optical Network Controller Infra APIs

Alongside the Cisco Optical Network Controller, a range of additional APIs is available, providing various functionalities for system configuration, management, and monitoring. These APIs offer tools for you to enhance their interaction with the network environment.

Key Capabilities:

1. **Authentication and Access**:
    - Features multiple methods for user authentication and authorization, supporting secure and controlled access to network resources.
2. **Configuration Management**:
    - Allows for configuration of system settings such as login banners and LDAP integration, enabling customization to meet specific operational needs.
3. **Performance Monitoring**:
    - Provides access to system performance metrics, offering insights into resource utilization and facilitating efficient system management.
4. **Log Management**:
    - Supports querying and managing logs to maintain visibility into network activities, assisting in effective monitoring and troubleshooting.
5. **User and Role Management**:
    - Enables the management of user accounts and roles, ensuring efficient administration of access and permissions within the network.

Accessing the APIs

You can access the APIs using the following common prefix:

For Authenticator API:

```
https://{CONC-HOST}:8443/nxf/api/v1/security
```

For Loki and Controller APIs:

```
https://{CONC-HOST}:8443/nxf/
```

CHAPTER 1

NxF Authenticator

The Authenticator API provides essential services for managing user authentication and authorization within the network environment. This API supports various authentication methods, including cookies, basic authentication, and token-based authentication. It also includes functionalities for handling user sessions, password changes, and single sign-on (SSO) configurations, ensuring secure and seamless access for users.

Accessing the APIs

You can access the Authenticator APIs using the following common prefix:

```
https://{CONC-HOST}:8443/nxf/api/v1/security
```

API methods: authentication

GET /changepass

Responses

200

Change password page

Media types

text/html

Security Requirements

POST /changepass

Request body

Required: true

Media types

application/x-www-form-urlencoded

Schema

Properties

username
string
password
string
newpassword
string
confirmnewpassword
string

type
object
required

- username
- password
- newpassword
- confirmnewpassword

Responses

301

Redirect to root

Security Requirements

API methods: config

GET /config/banner

Get login banner configuration

Responses

200

Login banner configuration

Media types

application/json

Schema

Properties

enabled
boolean
htmlString
string

array
enabled

500

Server error

Security Requirements

- Token
  - permission/admin

PUT /config/banner

Update login banner configuration

Request body

Required: true

Media types

application/json

Schema

Properties

enabled
boolean
htmlString
string

array
enabled

Responses

204

Config updated successfully

400

Config properties validation error

500

Server error

Security Requirements

- Token
  - permission/admin

GET /config/ldap

Get LDAP configuration

Responses

200

LDAP configuration

Media types

application/json

Schema

Properties

enabled
boolean
url
string
bindDn
string
hasCredentialsSet
boolean
searchBase
string
searchFilter
array

items

properties

attribute
string
value
string

required

- attribute
- value

groupsAttribute
string
rootCAs
string

array
enabled

500

Server error

Security Requirements

- Token
  - permission/admin

PUT /config/ldap

Update LDAP configuration

Request body

Required: true

Media types

application/json

Schema

Properties

enabled
boolean
url
string
bindDn
string
bindCredentials
string
searchBase
string
searchFilter
array

items

properties

attribute
string

value
string

required

- attribute
- value

groupsAttribute
string

rootCAs
string

array
enabled

Responses

204

Config updated successfully

400

Config properties validation error

500

Server error

Security Requirements

- Token
    - permission/admin

GET /config/limiter

Get login limiter configuration

Responses

200

Limiter configuration

Media types

application/json

Schema

Properties

windowMs
integer

minimum
60000

maxAttempts
integer

minimum
1

delayAfterAttempt
integer

minimum
1

delayMs
integer

minimum
100

500

Server error

Security Requirements

- Token
    - permission/admin

PUT /config/limiter

Update login limiter configuration

Request body

Required: true

Media types

application/json

Schema

Properties

windowMs
integer

minimum
60000

maxAttempts
integer

minimum
1

delayAfterAttempt
integer

minimum
1

delayMs
integer

minimum
100

Responses

204

Config updated successfully

400

Config properties validation error

500

Server error

Security Requirements

- Token
  - permission/admin

POST /config/mapping

Add permission mapping

Request body

Required: true

Media types

application/json

Schema

Properties

guid
string

type
string

enum

- LDAP_USER
- LDAP_GROUP
- SAML_USER
- SAML_GROUP

match
string

access
array

items
string

required

- type
- match
- access

Responses

201

Mapping added successfully

Headers

Name	Description	Schema
location	URL of newly created mapping	**type** string

500

Server error

Security Requirements

- Token
  - permission/admin

PUT /config/mapping/{guid}

Update permission mapping

Parameters

guid (path)

Mapping GUID

type
string

required
true

Request body

Required: true

Media types

application/json

Schema

Properties

guid
string

type
string

enum

- LDAP_USER
- LDAP_GROUP
- SAML_USER
- SAML_GROUP

match
string

access
array

items
string

required

- type
- match
- access

Responses

204

Mapping updated successfully

Headers

Name	Description	Schema
location	URL of updated mapping	**type** string

400

Validation error

500

Server error

Security Requirements

- Token
  - permission/admin

DELETE /config/mapping/{guid}

Delete permission mapping

Parameters

guid (path)

Mapping GUID

type
string

required
true

Responses

204

Mapping deleted successfully

400

Validation error

500

Server error

Security Requirements

- Token
  - permission/admin

GET /config/passwordPolicy

Get password policy configuration

Responses

200

Password policy configuration

Media types

application/json

Schema

Properties

passwordExpirationDays
integer

minimum
1

default
180

passwordReuseLimit
integer

minimum
3
maximum
24
default
12

minComplexityScore
integer

minimum
0
maximum
5
default
3

500

Server error

Security Requirements

- Token
  - permission/admin

PUT /config/passwordPolicy

Update password policy configuration

Request body

Required: true

Media types

application/json

Schema

Properties

passwordExpirationDays
integer

minimum
1
default
180

passwordReuseLimit
integer

minimum
3
maximum
24
default
12

minComplexityScore
integer

minimum
0
maximum
5
default
3

Responses

204

Config updated successfully

400

Config properties validation error

500

Server error

Security Requirements

- Token
  - permission/admin

GET /config/session

Get session configuration

Responses

200

Session configuration

Media types

application/json

Schema

Properties

cookieName
string

maxAge
integer

minimum
60

maximum
31536000

secure
boolean

rolling
boolean

sameSite
string

enum

- lax
- strict
- none

domain
string

500

Server error

Security Requirements

- Token
  - permission/admin

PUT /config/session

Update session configuration

Request body

Required: true

Media types

application/json

Schema

Properties

cookieName
string

maxAge
integer

minimum
60

maximum
31536000

secure
boolean

rolling
boolean

sameSite
string

enum

- lax
- strict
- none

domain
string

Responses

204

Config updated successfully

400

Config properties validation error

500

Server error

Security Requirements

- Token
  - permission/admin

GET /config/sso

Get SAML configuration

Responses

200

SAML configuration

Media types

application/json

Schema

Properties

enabled
boolean
loginUrl
string
groupsAttribute
string
entityId
string
baseUrl
string
signingCertificate
string

array
enabled

500

Server error

Security Requirements

- Token
  - permission/admin

PUT /config/sso

Update SAML configuration

Request body

Required: true

Media types

application/json

Schema

Properties

enabled
boolean

loginUrl
string
groupsAttribute
string
entityId
string
baseUrl
string
signingCertificate
string

array
enabled

Responses

204

Config updated successfully

400

Config properties validation error

500

Server error

Security Requirements

- Token
    - permission/admin

API methods: mapping

GET /config/mapping

Get all permission mappings

Responses

200

Array of all permission mappings

Media types

application/json

Schema

type
array

items

properties

stringLDAP_USERLDAP_GROUPSAML_USERSAML_GROUP

guid

string

match

string

access

array

items

string

required

- type
- match
- access

500

Server error

Security Requirements

- Token
  - permission/admin

API methods: login

GET /login

Responses

200

Login page

Media types

text/html

POST /login

Request body

Required: true

Media types

application/x-www-form-urlencoded

Schema

Properties

username
string
password
string
_csrf
string
submit
string

type
object
required

- username
- password
- _csrf
- submit

Responses

301

Redirect to root after successful login, /login on failed or /changepass if password expired

Security Requirements

API methods: users

GET /user

Get all local users

Responses

200

Local users array

Media types

application/json

Schema

type
array

items

properties

guid
string
username
string
access
array

items
string

active
boolean
locked
boolean
created
string

readOnly
true

updated
string

readOnly
true

expires

anyOf

- string
- string

enum

- never
- default

mustChangePassword
boolean
displayName
string
desc
string

required

- expires
- username
- access
- active
- locked

500

Server error

Security Requirements

- Token
    - permission/admin

POST /user

Create local user

Request body

Required: true

Media types

application/json

Schema

allOf

- **properties**

  guid
  string
  username
  string
  access
  array

  items
  string

  active
  boolean
  locked
  boolean
  created
  string

  readOnly
  true

  updated
  string

  readOnly
  true

  expires

  anyOf

  - string

- string

  enum

  - never
  - default

mustChangePassword
boolean

displayName
string

desc
string

required

- expires
- username
- access
- active
- locked

- object

  properties

  password
  string

Example

```
{
  "username": "user1",
  "access": ["permission/admin"],
  "active": true,
  "locked": false,
  "displayName": "User 1",
  "desc": "User 1 description",
  "password": 123456
}
```

Responses

201

User created successfully

500

Server error

Security Requirements

- Token
  - permission/admin

PUT /user/{guid}

Update local user

Parameters

guid (path)

User GUID

type
string

required
true

Request body

Required: true

Media types

application/json

Schema

allOf

- **properties**

  guid
  string
  username
  string
  access
  array

  items
  string

  active
  boolean
  locked
  boolean
  created
  string

  readOnly
  true

  updated
  string

  readOnly
  true

expires

anyOf

- string
- string

enum

- never
- default

mustChangePassword
boolean

displayName
string

desc
string

required

- expires
- username
- access
- active
- locked

- object

properties

password
string

Example

```
{
  "username": "user1",
  "access": ["permission/admin"],
  "active": true,
  "locked": false,
  "displayName": "User 1",
  "desc": "User 1 description",
  "password": 123456
}
```

Responses

204

Change applied successfuly

500

Server error

Security Requirements

- Token
    - permission/admin

DELETE /user/{guid}

Delete local user

Parameters

guid (path)

User GUID

type
string

required
true

Responses

204

User deleted successfully

500

Server error

Security Requirements

- Token
  - permission/admin

Components

Schemas

LdapProperties

Properties

enabled
boolean
url
string
bindDn
string
bindCredentials
string
searchBase
string
searchFilter
array

items

properties

attribute
string
value
string

required

- attribute
- value

groupsAttribute
string
rootCAs
string

array
enabled

LdapSearchFilter

Properties

attribute
string
value
string

required

- attribute
- value

LoginBannerProperties

Properties

enabled
boolean
htmlString
string

array
enabled

LoginLimiterProperties

Properties

windowMs
integer

minimum
60000

maxAttempts
integer

minimum
1

delayAfterAttempt
integer

minimum
1

delayMs
integer

minimum
100

PasswordPolicyProperties

Properties

passwordExpirationDays
integer

minimum
1
default
180

passwordReuseLimit
integer

minimum
3
maximum
24
default
12

minComplexityScore
integer

minimum
0
maximum
5
default
3

PermissionMapping

Properties

guid
string
type
string

enum

- LDAP_USER
- LDAP_GROUP
- SAML_USER
- SAML_GROUP

match
string
access
array

items
string

required

- type
- match
- access

PublicLdapProperties

Properties

enabled
boolean
url
string
bindDn
string
hasCredentialsSet
boolean
searchBase
string
searchFilter
array

items

properties

attribute
string
value
string

required

- attribute
- value

groupsAttribute
string
rootCAs
string

array
enabled

SamlProperties

Properties

- **enabled**
  boolean
- **loginUrl**
  string
- **groupsAttribute**
  string
- **entityId**
  string
- **baseUrl**
  string
- **signingCertificate**
  string

array
enabled

SessionProperties

Properties

- **cookieName**
  string
- **maxAge**
  integer
  - **minimum**
    60
  - **maximum**
    31536000
- **secure**
  boolean
- **rolling**
  boolean
- **sameSite**
  string
  - **enum**
    - lax
    - strict
    - none
- **domain**
  string

User

Properties

- **guid**
  string
- **username**
  string
- **access**
  array
  - **items**
    string
- **active**
  boolean
- **locked**
  boolean
- **created**
  string
  - **readOnly**
    true
- **updated**
  string
  - **readOnly**
    true
- **expires**
  - **anyOf**
    - string
    - string
      - **enum**
        - never
        - default
- **mustChangePassword**
  boolean
- **displayName**
  string
- **desc**
  string

required

- expires
- username
- access
- active
- locked

UserCreationProperties

allOf

- **properties**

  guid
  string
  username
  string
  access
  array

  items
  string

  active
  boolean
  locked
  boolean
  created
  string

  readOnly
  true

  updated
  string

  readOnly
  true

  expires

  anyOf

  - string
  - string

  enum

  - never
  - default

  mustChangePassword
  boolean
  displayName
  string
  desc
  string

  required

  - expires
  - username
  - access
  - active
  - locked

- object

  properties

  password
  string

Security schemes

Cookie

Name
_nxf_auth
Type
apiKey
Location
cookie

BasicAuth

Type
http
HTTP Authorization scheme
basic

Token

Name
X-NXF-Token
Type
apiKey
Location
header

CHAPTER 2

NxF Controller

Version: **3.1-463**

The Controller API offers a set of tools for managing system-level operations and configurations. It includes endpoints for retrieving and updating system metrics, roles, and applications. This API facilitates comprehensive system monitoring and management, enabling administrators to maintain optimal performance and resource allocation within the network infrastructure.

Servers

`https://{server}:8443/nxf/`

External Access

Table 1: Server variables

Name	Description
server	Remote server IP/Hostname, for example: nxf.cisco.com **Default:** localhost

Accessing the APIs

You can access the Controller APIs using the following common prefix:

```
https://{CONC-HOST}:8443/nxf/
```

API methods: system

GET /api/v1/apps

Responses

200

Installed apps

Media types

application/json

Schema

type
array
items

properties

name
string
baseUrl
string
enabled
boolean
default
boolean
icon
string

Security Requirements

- Token
    - permission/*

GET /api/v1/metrics/containers

Responses

200

Container metrics

Media types

application/json

Schema

type
array

items

properties

namespace
string
pod
string
container
string
timestamp
number
cpuSecs
number
cpuUsage
number
memWorkingBytes
number
fsAvailableBytes
number
fsUsedBytes
number

Security Requirements

- Token
  - permission/admin

GET /api/v1/metrics/nodes

Responses

200

Nodes metrics

Media types

application/json

Schema

type
object
patternProperties

.*

properties

nodeName
string
timestamp
number

load1m
number
load5m
number
load15m
number
cpuTotalCores
number
memFreeBytes
number
memTotalBytes
number
memAvailableBytes
number
disk
array

items

properties

mount
string
totalSizeBytes
number
availSizeBytes
number

Security Requirements

- Token
    - permission/admin

GET /api/v1/version

Responses

200

System versions

Media types

application/json

Schema

Properties

nextfusion
string
images
array

items

properties

name
string
version
string
size
integer
hash
string
nodes
array

items
string

type
object

Security Requirements

- Token
    - permission/*

Components

Schemas

App

Properties

name
string
baseUrl
string
enabled
boolean
default
boolean
icon
string

ContainerStat

Properties

namespace
string

pod
: string

container
: string

timestamp
: number

cpuSecs
: number

cpuUsage
: number

memWorkingBytes
: number

fsAvailableBytes
: number

fsUsedBytes
: number

DiskStatus

Properties

mount
: string

totalSizeBytes
: number

availSizeBytes
: number

Image

Properties

name
: string

version
: string

size
: integer

hash
: string

nodes
: array

  items
  : string

NodeStat

Properties

nodeName
: string

timestamp
: number

load1m
number
load5m
number
load15m
number
cpuTotalCores
number
memFreeBytes
number
memTotalBytes
number
memAvailableBytes
number
disk
array

items

properties

mount
string
totalSizeBytes
number
availSizeBytes
number

Role

Properties

name
string
access
array

items
string

Security schemes

Token

Name
X-NXF-Token
Type
apiKey
Location
header

CHAPTER 3

NxF Loki

Version: **1.1**

The Loki API is designed for efficient log management and retrieval. It provides functionalities to push log entries, query logs, and manage log data efficiently. With this API, users can perform complex log queries, manage log streams, and ensure that logging data is accessible and organized for analysis and auditing purposes.

Servers

`https://{server}:8443/nxf/api/v1/loki`

External Access

Table 2: Server variables

Name	Description
server	Remote server IP/Hostname, for example: nxf.cisco.com **Default:** localhost

Accessing the APIs

You can access the Loki APIs using the following common prefix:

```
https://{CONC-HOST}:8443/nxf/
```

- API methods: loki, on page 41
- Components, on page 45

API methods: loki

GET /api/v1/query

Parameters

query (path)

LogQL query to perform

type
string

required
true

limit (path)

Max number of entries to return

type
number

default
100

time (path)

Evaluation time for the query as a nanosecond Unix epoch

type
number

direction (path)

Determines the sort order of logs

type
string

enum

- forward
- backward

Responses

200

Query was successfull

Media types

application/json

Schema

Properties

status
text

data
object

properties

result
array

items

properties

stream
object
values
array

items
array

prefixItems

- number
- string

type
object

Security Requirements

- Token
  - permission/admin

GET /api/v1/query_range

Parameters

query (path)

LogQL query to perform

type
string

required
true

limit (path)

Max number of entries to return

type
number
default
100

start (path)

Start time for the query as a nanosecond Unix epoch

type
number

end (path)

End time for the query as a nanosecond Unix epoch

type
number

since (path)

A duration used to calculate start relative to end

type
number

step (path)

Query resolution step width in duration format or float number of seconds

type
string

interval (path)

Only return entries at (or greater than) the specified interval

type
number

direction (path)

Determines the sort order of logs

type
string

enum

- forward
- backward

Responses

200

Query was successfull

Media types

application/json

Schema

Properties

status
text

data
object

properties

result
array

items

properties

stream
object

values
array

items
array

prefixItems

- number
- string

type
object

Security Requirements

- Token
  - permission/admin

Components

Schemas

StreamEntry

Properties

stream
object

values
array

items
array

prefixItems

- number
- string

Security schemes

Token

Name
X-NXF-Token

Type
: apiKey

Location
: header