



Manage NVMe subsystems

REST API reference

NetApp
September 09, 2025

This PDF was generated from https://docs.netapp.com/us-en/ontap-restapi-9161/protocols_nvme_subsystems_endpoint_overview.html on September 09, 2025. Always check docs.netapp.com for the latest.

Table of Contents

Manage NVMe subsystems	1
Protocols NVMe subsystems endpoint overview	1
Overview	1
Examples	1
An NVMe Subsystem Host	7
Examples	7
Retrieve NVMe subsystems	9
Related ONTAP commands	9
Learn more	10
Parameters	10
Response	13
Error	15
Definitions	16
Create an NVMe subsystem	24
Required properties	24
Related ONTAP commands	24
Learn more	24
Parameters	24
Request Body	24
Response	28
Error	30
Definitions	32
Retrieve NVMe subsystem hosts	42
Expensive properties	42
Related ONTAP commands	42
Learn more	42
Parameters	42
Response	43
Error	45
Definitions	45
Add NVMe subsystem hosts	51
Required properties	51
Related ONTAP commands	51
Learn more	51
Parameters	52
Request Body	52
Response	55
Error	57
Definitions	58
Delete an NVMe subsystem host	66
Related ONTAP commands	66
Learn more	66
Parameters	67

Response	67
Error	67
Definitions	68
Retrieve an NVMe subsystem host	69
Related ONTAP commands	69
Learn more	69
Parameters	69
Response	69
Error	71
Definitions	72
Remove an NVMe subsystem	77
Related ONTAP commands	78
Learn more	78
Parameters	78
Response	78
Error	78
Definitions	79
Retrieve an NVMe subsystem	80
Expensive properties	80
Related ONTAP commands	80
Learn more	81
Parameters	81
Response	81
Error	85
Definitions	86
Update an NVMe subsystem	92
Related ONTAP commands	92
Learn more	92
Parameters	92
Request Body	92
Response	94
Error	94
Definitions	95

Manage NVMe subsystems

Protocols NVMe subsystems endpoint overview

Overview

An NVMe subsystem maintains configuration state and namespace access control for a set of NVMe-connected hosts.

The NVMe subsystem REST API allows you to create, update, delete, and discover NVMe subsystems as well as add and remove NVMe hosts that can access the subsystem and associated namespaces.

Examples

Creating an NVMe subsystem

```
# The API:
POST /api/protocols/nvme/subsystems

# The call:
curl -X POST 'https://<mgmt-ip>/api/protocols/nvme/subsystems' -H 'Accept: application/json' -d '{ "svm": { "name": "svm1" }, "name": "subsystem1", "os_type": "linux" }'
```

Creating an NVMe subsystem with multiple NVMe subsystem hosts

```
# The API:
POST /api/protocols/nvme/subsystems

# The call:
curl -X POST 'https://<mgmt-ip>/api/protocols/nvme/subsystems' -H 'Accept: application/json' -d '{ "svm": { "name": "svm1" }, "name": "subsystem2", "os_type": "vmware", "hosts": [ { "nqn": "nqn.1992-01.example.com:host1" }, { "nqn": "nqn.1992-01.example.com:host2" } ] }'
```

Retrieving all NVMe subsystems

```
# The API:
GET /api/protocols/nvme/subsystems

# The call:
curl -X GET 'https://<mgmt-ip>/api/protocols/nvme/subsystems' -H 'Accept:
application/json'

# The response:
{
  "records": [
    {
      "svm": {
        "uuid": "a009a9e7-4081-b576-7575-ada21efcaf16",
        "name": "svm1",
      },
      "uuid": "acde901a-a379-4a91-9ea6-1b728ed6696f",
      "name": "subsystem1",
    },
    {
      "svm": {
        "uuid": "a009a9e7-4081-b576-7575-ada21efcaf16",
        "name": "svm1",
      },
      "uuid": "bcde901a-a379-4a91-9ea6-1b728ed6696f",
      "name": "subsystem2",
    }
  ],
  "num_records": 2,
}
```

Retrieving all NVMe subsystems with OS type *linux*

Note that the `os_type` query parameter is used to perform the query.

```
# The API:
GET /api/protocols/nvme/subsystems

# The call:
curl -X GET 'https://<mgmt-
ip>/api/protocols/nvme/subsystems?os_type=linux' -H 'Accept:
application/json'

# The response:
{
  "records": [
    {
      "svm": {
        "uuid": "a009a9e7-4081-b576-7575-ada21efcaf16",
        "name": "svm1",
      },
      "uuid": "acde901a-a379-4a91-9ea6-1b728ed6696f",
      "name": "subsystem1",
      "os_type": "linux",
    }
  ],
  "num_records": 1,
}
```

Retrieving a specific NVMe subsystem

```
# The API:
GET /api/protocols/nvme/subsystems/{uuid}

# The call:
curl -X GET 'https://<mgmt-ip>/api/protocols/nvme/subsystems/acde901a-
a379-4a91-9ea6-1b728ed6696f' -H 'Accept: application/json'

# The response:
{
  "svm": {
    "uuid": "a009a9e7-4081-b576-7575-ada21efcaf16",
    "name": "svm1",
  },
  "uuid": "acde901a-a379-4a91-9ea6-1b728ed6696f",
  "name": "subsystem1",
  "os_type": "linux",
  "target_nqn": "nqn.1992-
08.com.netapp:sn.d04594ef915b4c73b642169e72e4c0b1:subsystem.subsystem1",
  "serial_number": "wtJNKNKD-uPLAAAAAAD",
  "io_queue": {
    "default": {
      "count": 4,
      "depth": 32
    }
  }
}
```

Retrieving the NVMe namespaces mapped to a specific NVMe subsystem

Note that the `fields` query parameter is used to specify the desired properties.

```
# The API:
GET /api/protocols/nvme/subsystems/{uuid}

# The call:
curl -X GET 'https://<mgmt-ip>/api/protocols/nvme/subsystems/acde901a-
a379-4a91-9ea6-1b728ed6696f?fields=subsystem_maps' -H 'Accept:
application/json'

# The response:
{
  "svm": {
    "uuid": "a009a9e7-4081-b576-7575-ada21efcaf16",
    "name": "svml",
  },
  "uuid": "acde901a-a379-4a91-9ea6-1b728ed6696f",
  "name": "subsystem1",
  "subsystem_maps": [
    {
      "anagrpid": "00000001h",
      "namespace": {
        "uuid": "eeaaca23-128d-4a7d-be4a-dc9106705799",
        "name": "/vol/vol1/namespace1"
      },
      "nsid": "00000001h"
    },
    {
      "anagrpid": "00000002h",
      "namespace": {
        "uuid": "feaaca23-83a0-4a7d-beda-dc9106705799",
        "name": "/vol/vol1/namespace2"
      },
      "nsid": "00000002h"
    }
  ]
}
```

Adding a comment about an NVMe subsystem

```
# The API:
PATCH /api/protocols/nvme/subsystems/{uuid}

# The call:
curl -X PATCH 'https://<mgmt-ip>/api/protocols/nvme/subsystems/acde901a-
a379-4a91-9ea6-1b728ed6696f' -H 'Accept: application/json' -d '{
"comment": "A brief comment about the subsystem" }'
```

Deleting an NVMe subsystem

```
# The API:
DELETE /api/protocols/nvme/subsystems/{uuid}

# The call:
curl -X DELETE 'https://<mgmt-ip>/api/protocols/nvme/subsystems/acde901a-
a379-4a91-9ea6-1b728ed6696f' -H 'Accept: application/json'
```

Deleting an NVMe subsystem with mapped NVMe namespaces

Normally, deleting an NVMe subsystem that has mapped NVMe namespaces is not allowed. The deletion can be forced using the `allow_delete_while_mapped` query parameter.

```
# The API:
DELETE /api/protocols/nvme/subsystems/{uuid}

# The call:
curl -X DELETE 'https://<mgmt-ip>/api/protocols/nvme/subsystems/acde901a-
a379-4a91-9ea6-1b728ed6696f?allow_delete_while_mapped=true' -H 'Accept:
application/json'
```

Delete an NVMe subsystem with NVMe subsystem hosts

Normally, deleting an NVMe subsystem with NVMe subsystem hosts is disallowed. The deletion can be forced using the `allow_delete_with_hosts` query parameter.

```
# The API:
DELETE /api/protocols/nvme/subsystems/{uuid}

# The call:
curl -X DELETE 'https://<mgmt-ip>/api/protocols/nvme/subsystems/acde901a-
a379-4a91-9ea6-1b728ed6696f?allow_delete_with_hosts=true' -H 'Accept:
application/json'
```

An NVMe Subsystem Host

An NVMe subsystem host is a network host provisioned to an NVMe subsystem to access namespaces mapped to that subsystem.

Examples

Adding an NVMe subsystem host to an NVMe subsystem

```
# The API:
POST /protocols/nvme/subsystems/{subsystem.uuid}/hosts

# The call:
curl -X POST 'https://<mgmt-ip>/api/protocols/nvme/subsystems/acde901a-
a379-4a91-9ea6-1b728ed6696f/hosts' -H 'Accept: application/json' -d '{
  "nqn": "nqn.1992-01.com.example:subsys1.host1" }'
```

Adding multiple NVMe subsystem hosts to an NVMe subsystem

```
# The API:
POST /protocols/nvme/subsystems/{subsystem.uuid}/hosts

# The call:
curl -X POST 'https://<mgmt-ip>/api/protocols/nvme/subsystems/acde901a-
a379-4a91-9ea6-1b728ed6696f/hosts' -H 'Accept: application/json' -d '{
  "records": [ { "nqn": "nqn.1992-01.com.example:subsys1.host2" }, { "nqn":
    "nqn.1992-01.com.example:subsys1.host3" } ] }'
```

Retrieving all NVMe subsystem hosts for an NVMe subsystem

```
# The API:
GET /protocols/nvme/subsystems/{subsystem.uuid}/hosts

# The call:
curl -X GET 'https://<mgmt-ip>/api/protocols/nvme/subsystems/acde901a-
a379-4a91-9ea6-1b728ed6696f/hosts' -H 'Accept: application/json'

# The response:
{
  "records": [
    {
      "nqn": "nqn.1992-01.com.example:subsys1.host1",
    },
    {
      "nqn": "nqn.1992-01.com.example:subsys1.host2",
    },
    {
      "nqn": "nqn.1992-01.com.example:subsys1.host3",
    }
  ],
  "num_records": 3,
}
```

Retrieving a specific NVMe subsystem host for an NVMe subsystem

```
# The API:
GET /protocols/nvme/subsystems/{subsystem.uuid}/hosts/{nqn}

# The call:
curl -X GET 'https://<mgmt-ip>/api/protocols/nvme/subsystems/acde901a-
a379-4a91-9ea6-1b728ed6696f/hosts/nqn.1992-01.com.example:subsys1.host1'
-H 'Accept: application/json'

# The response:
{
  "subsystem": {
    "uuid": "acde901a-a379-4a91-9ea6-1b728ed6696f",
  },
  "nqn": "nqn.1992-01.com.example:subsys1.host1",
  "dh_hmac_chap": {
    "node": "none"
  },
  "priority": "regular",
}
```

Deleting an NVMe subsystem host from an NVMe subsystem

```
# The API:
DELETE /protocols/nvme/subsystems/{subsystem.uuid}/hosts/{nqn}

# The call:
curl -X DELETE 'https://<mgmt-ip>/api/protocols/nvme/subsystems/acde901a-
a379-4a91-9ea6-1b728ed6696f/hosts/nqn.1992-01.com.example:subsys1.host1'
-H 'Accept: application/json'
```

Retrieve NVMe subsystems

GET /protocols/nvme/subsystems

Introduced In: 9.6

Retrieves NVMe subsystems.

Related ONTAP commands

- `vserver nvme subsystem host show`
- `vserver nvme subsystem map show`

- `vserver nvme subsystem show`

Learn more

- [DOC /protocols/nvme/subsystems](#)

Parameters

Name	Type	In	Required	Description
hosts.tls.key_type	string	query	False	Filter by hosts.tls.key_type • Introduced in: 9.16
hosts.priority	string	query	False	Filter by hosts.priority • Introduced in: 9.14
hosts.nqn	string	query	False	Filter by hosts.nqn
hosts.dh_hmac_chap.group_size	string	query	False	Filter by hosts.dh_hmac_chap.group_size • Introduced in: 9.12
hosts.dh_hmac_chap.mode	string	query	False	Filter by hosts.dh_hmac_chap.mode • Introduced in: 9.12
hosts.dh_hmac_chap.hash_function	string	query	False	Filter by hosts.dh_hmac_chap.hash_function • Introduced in: 9.12
vendor_uuids	string	query	False	Filter by vendor_uuids • Introduced in: 9.9

Name	Type	In	Required	Description
name	string	query	False	Filter by name • maxLength: 64 • minLength: 1
serial_number	string	query	False	Filter by serial_number • maxLength: 20 • minLength: 20
svm.name	string	query	False	Filter by svm.name
svm.uuid	string	query	False	Filter by svm.uuid
delete_on_unmap	boolean	query	False	Filter by delete_on_unmap • Introduced in: 9.7
comment	string	query	False	Filter by comment • maxLength: 255 • minLength: 0
subsystem_maps.nsid	string	query	False	Filter by subsystem_maps.nsid
subsystem_maps.agrp_id	string	query	False	Filter by subsystem_maps.agrp_id
subsystem_maps.namespace.name	string	query	False	Filter by subsystem_maps.namespace.name
subsystem_maps.namespace.uuid	string	query	False	Filter by subsystem_maps.namespace.uuid
target_nqn	string	query	False	Filter by target_nqn • maxLength: 223 • minLength: 1

Name	Type	In	Required	Description
os_type	string	query	False	Filter by os_type
io_queue.default.count	integer	query	False	Filter by io_queue.default.count • Max value: 15 • Min value: 1
io_queue.default.depth	integer	query	False	Filter by io_queue.default.depth • Max value: 128 • Min value: 16
uuid	string	query	False	Filter by uuid
fields	array[string]	query	False	Specify the fields to return.
max_records	integer	query	False	Limit the number of records returned.
return_records	boolean	query	False	The default is true for GET calls. When set to false, only the number of records is returned. • Default value: 1

Name	Type	In	Required	Description
return_timeout	integer	query	False	The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached. • Max value: 120 • Min value: 0 • Default value: 1
order_by	array[string]	query	False	Order results by specified fields and optional [asc

Response

Status: 200, Ok

Name	Type	Description
_links	_links	
num_records	integer	The number of records in the response.
records	array[nvme_subsystem]	

Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": 1,
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "comment": "string",
      "hosts": [
        {
          "_links": {
            "self": {
              "href": "/api/resourcelink"
            }
          },
          "dh_hmac_chap": {
            "group_size": "string",
            "hash_function": "string",
            "mode": "bidirectional"
          },
          "nqn": "nqn.1992-01.example.com:string",
          "priority": "string",
          "tls": {
            "key_type": "configured"
          }
        }
      ],
      "io_queue": {
        "default": {
          "count": 4,
          "depth": 16
        }
      },
      "name": "subsystem1",
    }
  ]
}
```

```

"os_type": "string",
"serial_number": "wCVsgFMiuMhVAAAAAAB",
"subsystem_maps": [
  {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "anagrpid": "00103050h",
    "namespace": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "/vol/vol1/namespace1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "nsid": "00000001h"
  }
],
"svm": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "svm1",
  "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
},
"target_nqn": "nqn.1992-01.example.com:string",
"uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",
"vendor_uuids": [
  "1447f0f4-42e5-0dfc-871a-dc9b3f92d8f8"
]
}
]
}

```

Error

Status: Default, Error

Name	Type	Description
error	returned_error	

Example error

```
{
  "error": {
    "arguments": [
      {
        "code": "string",
        "message": "string"
      }
    ],
    "code": "4",
    "message": "entry doesn't exist",
    "target": "uuid"
  }
}
```

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

_links

Name	Type	Description
self	href	

dh_hmac_chap

A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.

Name	Type	Description
group_size	string	The Diffie-Hellman group size for NVMe in-band authentication. When property <code>host_secret_key</code> is provided, this property defaults to <code>2048_bit</code> . When supplied, the property <code>host_secret_key</code> must also be supplied. Optional in POST.
hash_function	string	The hash function for NVMe in-band authentication. When property <code>host_secret_key</code> is provided, this property defaults to <code>sha_256</code> . When supplied, the property <code>host_secret_key</code> must also be supplied. Optional in POST.

Name	Type	Description
mode	string	The expected NVMe in-band authentication mode for the host. This property is an indication of which secrets are configured for the host. When set to: • none: The host has neither the host nor controller secret configured, and no authentication is performed. • unidirectional: The host has a host secret configured. The controller will authenticate the host. • bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.

tls

A container for the configuration for NVMe/TCP-TLS transport session for the host.

Name	Type	Description
key_type	string	The method by which the TLS pre-shared key (PSK) is configured for the host. Optional in POST. The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree. Possible values: • <code>none</code> - TLS is not configured for the host connection. No value is allowed for property <code>configured_psk</code>. • <code>configured</code> - A user supplied PSK is configured for the NVMe/TCP-TLS transport connection between the host and the NVMe subsystem. A valid value for property <code>configured_psk</code> is required. This property defaults to <code>none</code> unless a value is supplied for <code>configured_psk</code> in which case it defaults to <code>configured</code>.

hosts

Name	Type	Description
_links	_links	
dh_hmac_chap	dh_hmac_chap	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.
nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target.

Name	Type	Description
priority	string	The host priority setting allocates appropriate NVMe I/O queues (count and depth) for the host to submit I/O commands. Absence of this property in GET implies user configured values of I/O queue count and I/O queue depth are being used.
tls	tls	A container for the configuration for NVMe/TCP-TLS transport session for the host.

default

The default I/O queue parameters inherited by NVMe hosts in the NVMe subsystem.

Name	Type	Description
count	integer	The number of host I/O queue pairs.
depth	integer	The host I/O queue depth.

io_queue

The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.

Name	Type	Description
default	default	The default I/O queue parameters inherited by NVMe hosts in the NVMe subsystem.

namespace

An NVMe namespace mapped to the NVMe subsystem.

Name	Type	Description
_links	_links	
name	string	The name of the NVMe namespace.
uuid	string	The unique identifier of the NVMe namespace.

subsystem_maps

An NVMe namespace mapped to the NVMe subsystem.

Name	Type	Description
_links	_links	
anagrpId	string	The Asymmetric Namespace Access Group ID (ANAGRPID) of the NVMe namespace. The format for an ANAGRPI is 8 hexadecimal digits (zero-filled) followed by a lower case "h".
namespace	namespace	An NVMe namespace mapped to the NVMe subsystem.
nsid	string	The NVMe namespace identifier. This is an identifier used by an NVMe controller to provide access to the NVMe namespace. The format for an NVMe namespace identifier is 8 hexadecimal digits (zero-filled) followed by a lower case "h".

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
_links	_links	
name	string	The name of the SVM. This field cannot be specified in a PATCH method.
uuid	string	The unique identifier of the SVM. This field cannot be specified in a PATCH method.

nvme_subsystem

An NVMe subsystem maintains configuration state and namespace access control for a set of NVMe-connected hosts.

Name	Type	Description
_links	_links	

Name	Type	Description
comment	string	A configurable comment for the NVMe subsystem. Optional in POST and PATCH.
delete_on_unmap	boolean	An option that causes the subsystem to be deleted when the last subsystem map associated with it is deleted. Optional in POST and PATCH. This property defaults to <i>false</i> when the subsystem is created.
hosts	array[hosts]	The NVMe hosts configured for access to the NVMe subsystem. Optional in POST.
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.
name	string	The name of the NVMe subsystem. Once created, an NVMe subsystem cannot be renamed. Required in POST.
os_type	string	The host operating system of the NVMe subsystem's hosts. Required in POST.
serial_number	string	The serial number of the NVMe subsystem.
subsystem_maps	array[subsystem_maps]	The NVMe namespaces mapped to the NVMe subsystem. There is an added computational cost to retrieving property values for <code>subsystem_maps</code>. They are not populated for a GET request unless explicitly requested using the <code>fields</code> query parameter. See Requesting specific fields to learn more.
svm	svm	SVM, applies only to SVM-scoped objects.

Name	Type	Description
target_nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target.
uuid	string	The unique identifier of the NVMe subsystem.
vendor_uuids	array[string]	Vendor-specific identifiers (UUIDs) optionally assigned to an NVMe subsystem when the subsystem is created. The identifiers are used to enable vendor-specific NVMe protocol features. The identifiers are provided by a host application vendor and shared with NetApp prior to a joint product release. Creating an NVMe subsystem with an unknown or non-specific identifier will have no effect on the NVMe subsystem. Refer to the ONTAP SAN Administration Guide for a list of the supported vendor-specific identifiers. After a subsystem is created, the vendor-specific identifiers cannot be changed or removed. Optional in POST. • Introduced in: 9.9 • readCreate: 1

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

returned_error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code

Name	Type	Description
message	string	Error message
target	string	The target parameter that caused the error.

Create an NVMe subsystem

POST /protocols/nvme/subsystems

Introduced In: 9.6

Creates an NVMe subsystem.

Required properties

- `svm.uuid` or `svm.name` - Existing SVM in which to create the NVMe subsystem.
- `name` - Name for NVMe subsystem. Once created, an NVMe subsystem cannot be renamed.
- `os_type` - Operating system of the NVMe subsystem's hosts.

Related ONTAP commands

- `vserver nvme subsystem create`

Learn more

- [DOC /protocols/nvme/subsystems](#)

Parameters

Name	Type	In	Required	Description
return_records	boolean	query	False	The default is false. If set to true, the records are returned. • Default value:

Request Body

Name	Type	Description
comment	string	A configurable comment for the NVMe subsystem. Optional in POST and PATCH.

Name	Type	Description
delete_on_unmap	boolean	An option that causes the subsystem to be deleted when the last subsystem map associated with it is deleted. Optional in POST and PATCH. This property defaults to <i>false</i> when the subsystem is created.
hosts	array[hosts]	The NVMe hosts configured for access to the NVMe subsystem. Optional in POST.
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.
name	string	The name of the NVMe subsystem. Once created, an NVMe subsystem cannot be renamed. Required in POST.
os_type	string	The host operating system of the NVMe subsystem's hosts. Required in POST.
serial_number	string	The serial number of the NVMe subsystem.
subsystem_maps	array[subsystem_maps]	The NVMe namespaces mapped to the NVMe subsystem. There is an added computational cost to retrieving property values for <code>subsystem_maps</code>. They are not populated for a GET request unless explicitly requested using the <code>fields</code> query parameter. See Requesting specific fields to learn more.
svm	svm	SVM, applies only to SVM-scoped objects.
target_nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target.

Name	Type	Description
uuid	string	The unique identifier of the NVMe subsystem.
vendor_uuids	array[string]	Vendor-specific identifiers (UUIDs) optionally assigned to an NVMe subsystem when the subsystem is created. The identifiers are used to enable vendor-specific NVMe protocol features. The identifiers are provided by a host application vendor and shared with NetApp prior to a joint product release. Creating an NVMe subsystem with an unknown or non-specific identifier will have no effect on the NVMe subsystem. Refer to the ONTAP SAN Administration Guide for a list of the supported vendor-specific identifiers. After a subsystem is created, the vendor-specific identifiers cannot be changed or removed. Optional in POST. • Introduced in: 9.9 • readCreate: 1

Example request

```
{
  "comment": "string",
  "hosts": [
    {
      "dh_hmac_chap": {
        "controller_secret_key": "DHHC-
1:00:ia6zGodOr4SEG0Zzaw398rpY0wqipUWj4jWjUh4HWUz6aQ2n:",
        "group_size": "string",
        "hash_function": "string",
        "host_secret_key": "DHHC-
1:00:ia6zGodOr4SEG0Zzaw398rpY0wqipUWj4jWjUh4HWUz6aQ2n:",
        "mode": "bidirectional"
      },
      "nqn": "nqn.1992-01.example.com:string",
      "priority": "string",
      "tls": {
        "configured_psk": "NVMeTLSkey-
1:01:VRLbtnN9AQb2WXW3c9+wEf/DRLz0QuLdbYvEhwtDwWnf9LrZ:",
        "key_type": "configured"
      }
    }
  ],
  "io_queue": {
    "default": {
      "count": 4,
      "depth": 16
    }
  },
  "name": "subsystem1",
  "os_type": "string",
  "serial_number": "wCVsgFMiuMhVAAAAAAB",
  "subsystem_maps": [
    {
      "anagrpid": "00103050h",
      "namespace": {
        "name": "/vol/vol1/namespace1",
        "uuid": "1cd8a442-86d1-11e0-aelc-123478563412"
      },
      "nsid": "00000001h"
    }
  ],
  "svm": {
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
```

```
},
"target_nqn": "nqn.1992-01.example.com:string",
"uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",
"vendor_uuids": [
  "1447f0f4-42e5-0dfc-871a-dc9b3f92d8f8"
]
}
```

Response

Status: 201, Created

Name	Type	Description
num_records	integer	The number of records in the response.
records	array[nvme_subsystem]	

Example response

```
{
  "num_records": 1,
  "records": [
    {
      "comment": "string",
      "hosts": [
        {
          "dh_hmac_chap": {
            "controller_secret_key": "DHHC-
1:00:ia6zGodOr4SEG0Zzaw398rpY0wqipUWj4jWjUh4HWUz6aQ2n:",
            "group_size": "string",
            "hash_function": "string",
            "host_secret_key": "DHHC-
1:00:ia6zGodOr4SEG0Zzaw398rpY0wqipUWj4jWjUh4HWUz6aQ2n:",
            "mode": "bidirectional"
          },
          "nqn": "nqn.1992-01.example.com:string",
          "priority": "string",
          "tls": {
            "configured_psk": "NVMeTLSkey-
1:01:VRLbtnN9AQb2WXW3c9+wEf/DRLz0QuLdbYvEhwtdWwNf9LrZ:",
            "key_type": "configured"
          }
        }
      ],
      "io_queue": {
        "default": {
          "count": 4,
          "depth": 16
        }
      },
      "name": "subsystem1",
      "os_type": "string",
      "serial_number": "wCVsgFMiuMhVAAAAAAB",
      "subsystem_maps": [
        {
          "anagrpid": "00103050h",
          "namespace": {
            "name": "/vol/vol1/namespace1",
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
          },
          "nsid": "00000001h"
        }
      ]
    }
  ],
}
```

```

    "svm": {
      "name": "svm1",
      "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
    },
    "target_nqn": "nqn.1992-01.example.com:string",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",
    "vendor_uuids": [
      "1447f0f4-42e5-0dfc-871a-dc9b3f92d8f8"
    ]
  }
]
}

```

Headers

Name	Description	Type
Location	Useful for tracking the resource location	string

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
2621462	The supplied SVM does not exist.
2621706	The specified <code>svm.uuid</code> and <code>svm.name</code> do not refer to the same SVM.
2621707	The <code>svm.uuid</code> or <code>svm.name</code> must be provided.
72089635	Setting vendor-specific UUIDs on NVMe subsystems is not supported until the effective cluster version is 9.9 or later.
72089636	Creating NVMe subsystems with <code>os_type</code> AIX is not supported until the effective cluster version is 9.13.1 or later.
72089709	The NVMe subsystem name contains an invalid character.
72089711	An invalid vendor-specific UUID was specified.
72089712	A duplicate vendor-specific UUID was specific.
72089713	Too many vendor UUIDs were supplied.

Error Code	Description
72089716	The DH-HMAC-CHAP secret property is invalid. DH-HMAC-CHAP secrets must be in the format "DHC-1:0X:<Base 64 encoded key and CRC>:", where X represents 0, 1, or 3 indicating no hash function, SHA-256, and SHA-512 respectively.
72089771	The NQN is invalid. A non-empty qualifier is required after the prefix. An example of a valid NQN is <i>nqn.1992-01.com.example:string</i> .
72089772	The NQN is invalid. Add the prefix ' <i>nqn</i> '. An example of a valid NQN is <i>nqn.1992-01.com.example:string</i> .
72089773	The NQN is invalid. The date field must be formatted <i>yyyy-mm</i> . An example of a valid NQN is <i>nqn.1992-01.com.example:string</i> .
72090003	A host to be added to an NVMe subsystem is missing the "nqn" property.
72090025	The NVMe subsystem already exists for the SVM.
72090029	The NVMe service does not exist.
72090030	A partial success occurred while adding multiple NVMe subsystem hosts to an NVMe subsystem.
72090036	An NVMe subsystem host NQN was duplicated in the input.
72090042	The DH-HMAC-CHAP secret property is required when setting any other NVMe in-band authentication properties for a host.
72090043	An igroup already exists with the requested NVMe subsystem name.
72090151	NVMe/TCP-TLS is not supported for the effective version of the cluster.
72090202	A provided NVMe subsystem host TLS configured PSK is not valid.
72090204	A TLS configured PSK was not provided when adding an NVMe subsystem host with the configured key type.
72090205	An invalid combination for the TLS key type and configured PSK values was provided when adding an NVMe subsystem host. When key type is "none", no configured PSK is allowed. When key type is "configured", a configured PSK is required.

Also see the table of common errors in the [Response body](#) overview section of this documentation.

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

dh_hmac_chap

A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.

Name	Type	Description
controller_secret_key	string	The controller secret for NVMe in-band authentication. The value of this property is used by the NVMe host to authenticate the NVMe controller while establishing a connection. If unset, the controller is not authenticated. When supplied, the property <code>host_secret_key</code> must also be supplied. Optional in POST. This property is write-only. The <code>mode</code> property can be used to identify if a controller secret has been set for the host, but the controller secret value cannot be read. To change the value, the host must be deleted from the subsystem and re-added.
group_size	string	The Diffie-Hellman group size for NVMe in-band authentication. When property <code>host_secret_key</code> is provided, this property defaults to <code>2048_bit</code>. When supplied, the property <code>host_secret_key</code> must also be supplied. Optional in POST.

Name	Type	Description
hash_function	string	The hash function for NVMe in-band authentication. When property <code>host_secret_key</code> is provided, this property defaults to <code>sha_256</code> . When supplied, the property <code>host_secret_key</code> must also be supplied. Optional in POST.
host_secret_key	string	The host secret for NVMe in-band authentication. The value of this property is used by the NVMe controller to authenticate the NVMe host while establishing a connection. If unset, no authentication is performed by the host or controller. This property must be supplied if any other NVMe in-band authentication properties are supplied. Optional in POST. This property is write-only. The <code>mode</code> property can be used to identify if a host secret has been set for the host, but the host secret value cannot be read. To change the value, the host must be deleted from the subsystem and re-added.

Name	Type	Description
mode	string	The expected NVMe in-band authentication mode for the host. This property is an indication of which secrets are configured for the host. When set to: • none: The host has neither the host nor controller secret configured, and no authentication is performed. • unidirectional: The host has a host secret configured. The controller will authenticate the host. • bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.

tls

A container for the configuration for NVMe/TCP-TLS transport session for the host.

Name	Type	Description
configured_psk	string	A user supplied pre-shared key (PSK) value in PSK Interchange Format. Optional in POST. The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree. This property is only allowed when <code>key_type</code> is configured. If <code>configured_psk</code> is supplied and <code>key_type</code> is unset, <code>key_type</code> defaults to <code>configured</code>. This property is write-only. The <code>key_type</code> property can be used to identify if a configured PSK has been set for the host, but the PSK value cannot be read. To change the value, the host must be deleted from the subsystem and re-added.

Name	Type	Description
key_type	string	The method by which the TLS pre-shared key (PSK) is configured for the host. Optional in POST. The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree. Possible values: • <code>none</code> - TLS is not configured for the host connection. No value is allowed for property <code>configured_psk</code>. • <code>configured</code> - A user supplied PSK is configured for the NVMe/TCP-TLS transport connection between the host and the NVMe subsystem. A valid value for property <code>configured_psk</code> is required. This property defaults to <code>none</code> unless a value is supplied for <code>configured_psk</code> in which case it defaults to <code>configured</code>.

hosts

Name	Type	Description
dh_hmac_chap	dh_hmac_chap	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.
nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target.

Name	Type	Description
priority	string	The host priority setting allocates appropriate NVMe I/O queues (count and depth) for the host to submit I/O commands. Absence of this property in GET implies user configured values of I/O queue count and I/O queue depth are being used.
tls	tls	A container for the configuration for NVMe/TCP-TLS transport session for the host.

default

The default I/O queue parameters inherited by NVMe hosts in the NVMe subsystem.

Name	Type	Description
count	integer	The number of host I/O queue pairs.
depth	integer	The host I/O queue depth.

io_queue

The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.

Name	Type	Description
default	default	The default I/O queue parameters inherited by NVMe hosts in the NVMe subsystem.

namespace

An NVMe namespace mapped to the NVMe subsystem.

Name	Type	Description
name	string	The name of the NVMe namespace.
uuid	string	The unique identifier of the NVMe namespace.

subsystem_maps

An NVMe namespace mapped to the NVMe subsystem.

Name	Type	Description
anagrpId	string	The Asymmetric Namespace Access Group ID (ANAGRPID) of the NVMe namespace. The format for an ANAGRPI is 8 hexadecimal digits (zero-filled) followed by a lower case "h".
namespace	namespace	An NVMe namespace mapped to the NVMe subsystem.
nsid	string	The NVMe namespace identifier. This is an identifier used by an NVMe controller to provide access to the NVMe namespace. The format for an NVMe namespace identifier is 8 hexadecimal digits (zero-filled) followed by a lower case "h".

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
name	string	The name of the SVM. This field cannot be specified in a PATCH method.
uuid	string	The unique identifier of the SVM. This field cannot be specified in a PATCH method.

nvme_subsystem

An NVMe subsystem maintains configuration state and namespace access control for a set of NVMe-connected hosts.

Name	Type	Description
comment	string	A configurable comment for the NVMe subsystem. Optional in POST and PATCH.

Name	Type	Description
delete_on_unmap	boolean	An option that causes the subsystem to be deleted when the last subsystem map associated with it is deleted. Optional in POST and PATCH. This property defaults to <i>false</i> when the subsystem is created.
hosts	array[hosts]	The NVMe hosts configured for access to the NVMe subsystem. Optional in POST.
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.
name	string	The name of the NVMe subsystem. Once created, an NVMe subsystem cannot be renamed. Required in POST.
os_type	string	The host operating system of the NVMe subsystem's hosts. Required in POST.
serial_number	string	The serial number of the NVMe subsystem.
subsystem_maps	array[subsystem_maps]	The NVMe namespaces mapped to the NVMe subsystem. There is an added computational cost to retrieving property values for <code>subsystem_maps</code>. They are not populated for a GET request unless explicitly requested using the <code>fields</code> query parameter. See Requesting specific fields to learn more.
svm	svm	SVM, applies only to SVM-scoped objects.
target_nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target.

Name	Type	Description
uuid	string	The unique identifier of the NVMe subsystem.
vendor_uuids	array[string]	Vendor-specific identifiers (UUIDs) optionally assigned to an NVMe subsystem when the subsystem is created. The identifiers are used to enable vendor-specific NVMe protocol features. The identifiers are provided by a host application vendor and shared with NetApp prior to a joint product release. Creating an NVMe subsystem with an unknown or non-specific identifier will have no effect on the NVMe subsystem. Refer to the ONTAP SAN Administration Guide for a list of the supported vendor-specific identifiers. After a subsystem is created, the vendor-specific identifiers cannot be changed or removed. Optional in POST. • Introduced in: 9.9 • readCreate: 1

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

returned_error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code
message	string	Error message

Name	Type	Description
target	string	The target parameter that caused the error.

Retrieve NVMe subsystem hosts

GET /protocols/nvme/subsystems/{subsystem.uuid}/hosts

Introduced In: 9.6

Retrieves the NVMe subsystem hosts of an NVMe subsystem.

Expensive properties

There is an added computational cost to retrieving values for these properties. They are not included by default in GET results and must be explicitly requested using the `fields` query parameter. See [Requesting specific fields](#) to learn more.

- `subsystem_maps.*`

Related ONTAP commands

- `vserver nvme subsystem map show`
- `vserver nvme subsystem show`

Learn more

- [DOC /protocols/nvme/subsystems](#)

Parameters

Name	Type	In	Required	Description
subsystem.uuid	string	path	True	The unique identifier of the NVMe subsystem.
fields	array[string]	query	False	Specify the fields to return.
max_records	integer	query	False	Limit the number of records returned.

Name	Type	In	Required	Description
return_records	boolean	query	False	The default is true for GET calls. When set to false, only the number of records is returned. • Default value: 1
return_timeout	integer	query	False	The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached. • Default value: 1 • Max value: 120 • Min value: 0
order_by	array[string]	query	False	Order results by specified fields and optional [asc

Response

Status: 200, Ok

Name	Type	Description
_links	_links	
num_records	integer	The number of records in the response.
records	array[nvme_subsystem_host]	

Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": 1,
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "dh_hmac_chap": {
        "group_size": "string",
        "hash_function": "string",
        "mode": "bidirectional"
      },
      "io_queue": {
        "count": 4,
        "depth": 32
      },
      "nqn": "nqn.1992-01.example.com:string",
      "priority": "string",
      "subsystem": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "subsystem1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      },
      "tls": {
        "key_type": "configured"
      }
    }
  ]
}
```

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
72090001	The NVMe subsystem does not exist.

Also see the table of common errors in the [Response body](#) overview section of this documentation.

Name	Type	Description
error	returned_error	

Example error

```
{
  "error": {
    "arguments": [
      {
        "code": "string",
        "message": "string"
      }
    ],
    "code": "4",
    "message": "entry doesn't exist",
    "target": "uuid"
  }
}
```

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

_links

Name	Type	Description
self	href	

dh_hmac_chap

A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.

Name	Type	Description
group_size	string	The Diffie-Hellman group size for NVMe in-band authentication. When property <code>host_secret_key</code> is provided, this property defaults to <code>2048_bit</code> . When supplied, the property <code>host_secret_key</code> must also be supplied. Optional in POST.
hash_function	string	The hash function for NVMe in-band authentication. When property <code>host_secret_key</code> is provided, this property defaults to <code>sha_256</code> . When supplied, the property <code>host_secret_key</code> must also be supplied. Optional in POST.

Name	Type	Description
mode	string	The expected NVMe in-band authentication mode for the host. This property is an indication of which secrets are configured for the host. When set to: • none: The host has neither the host nor controller secret configured, and no authentication is performed. • unidirectional: The host has a host secret configured. The controller will authenticate the host. • bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.

io_queue

The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.

Name	Type	Description
count	integer	The number of I/O queue pairs. Absence of this property in GET implies property priority is set and platform and transport protocol specific values for I/O queue count is being used. Valid in GET only.
depth	integer	The I/O queue depth. Absence of this property in GET implies property priority is set and platform and transport protocol specific values for I/O queue depth is being used. Valid in GET only.

io_queue

The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.

Name	Type	Description
count	integer	The number of I/O queue pairs. The default value is inherited from the owning NVMe subsystem.
depth	integer	The I/O queue depth. The default value is inherited from the owning NVMe subsystem.

subsystem

The NVMe subsystem to which the NVMe host has been provisioned.

Name	Type	Description
_links	_links	
name	string	The name of the NVMe subsystem.
uuid	string	The unique identifier of the NVMe subsystem.

tls

A container for the configuration for NVMe/TCP-TLS transport session for the host.

Name	Type	Description
key_type	string	The method by which the TLS pre-shared key (PSK) is configured for the host. Optional in POST. The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree. Possible values: • <code>none</code> - TLS is not configured for the host connection. No value is allowed for property <code>configured_psk</code>. • <code>configured</code> - A user supplied PSK is configured for the NVMe/TCP-TLS transport connection between the host and the NVMe subsystem. A valid value for property <code>configured_psk</code> is required. This property defaults to <code>none</code> unless a value is supplied for <code>configured_psk</code> in which case it defaults to <code>configured</code>.

records

The NVMe host provisioned to access NVMe namespaces mapped to a subsystem.

Name	Type	Description
<code>_links</code>	_links	
<code>dh_hmac_chap</code>	dh_hmac_chap	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.
<code>io_queue</code>	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.

Name	Type	Description
nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target. Not allowed in POST when the <code>records</code> property is used.
subsystem	subsystem	The NVMe subsystem to which the NVMe host has been provisioned.
tls	tls	A container for the configuration for NVMe/TCP-TLS transport session for the host.

nvme_subsystem_host

The NVMe host provisioned to access NVMe namespaces mapped to a subsystem.

Name	Type	Description
_links	_links	
dh_hmac_chap	dh_hmac_chap	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.
nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target.
priority	string	The host priority setting allocates appropriate NVMe I/O queues (count and depth) for the host to submit I/O commands. Absence of this property in GET implies <code>io_queue</code> count and I/O queue depth are being used.
subsystem	subsystem	The NVMe subsystem to which the NVMe host has been provisioned.

Name	Type	Description
tls	tls	A container for the configuration for NVMe/TCP-TLS transport session for the host.

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

returned_error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

Add NVMe subsystem hosts

POST /protocols/nvme/subsystems/{subsystem.uuid}/hosts

Introduced In: 9.6

Adds NVMe subsystem host(s) to an NVMe subsystem.

Required properties

- `nqn` or `records.nqn` - NVMe host(s) NQN(s) to add to the NVMe subsystem.

Related ONTAP commands

- `vserver nvme subsystem host add`

Learn more

- [DOC /protocols/nvme/subsystems](#)

Parameters

Name	Type	In	Required	Description
subsystem.uuid	string	path	True	The unique identifier of the NVMe subsystem.
return_records	boolean	query	False	The default is false. If set to true, the records are returned. • Default value:

Request Body

Name	Type	Description
dh_hmac_chap	dh_hmac_chap	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.
nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target.
priority	string	The host priority setting allocates appropriate NVMe I/O queues (count and depth) for the host to submit I/O commands. Absence of this property in GET implies io_queue count and I/O queue depth are being used.
records	array[records]	An array of NVMe hosts specified to add multiple NVMe hosts to an NVMe subsystem in a single API call. Valid in POST only.
subsystem	subsystem	The NVMe subsystem to which the NVMe host has been provisioned.

Name	Type	Description
tls	tls	A container for the configuration for NVMe/TCP-TLS transport session for the host.

Example request

```
{
  "dh_hmac_chap": {
    "controller_secret_key": "DHHC-
1:00:ia6zGodOr4SEG0Zzaw398rpY0wqipUWj4jWjUh4HWUz6aQ2n:",
    "group_size": "string",
    "hash_function": "string",
    "host_secret_key": "DHHC-
1:00:ia6zGodOr4SEG0Zzaw398rpY0wqipUWj4jWjUh4HWUz6aQ2n:",
    "mode": "bidirectional"
  },
  "io_queue": {
    "count": 4,
    "depth": 32
  },
  "nqn": "nqn.1992-01.example.com:string",
  "priority": "string",
  "records": [
    {
      "dh_hmac_chap": {
        "controller_secret_key": "DHHC-
1:00:ia6zGodOr4SEG0Zzaw398rpY0wqipUWj4jWjUh4HWUz6aQ2n:",
        "group_size": "string",
        "hash_function": "string",
        "host_secret_key": "DHHC-
1:00:ia6zGodOr4SEG0Zzaw398rpY0wqipUWj4jWjUh4HWUz6aQ2n:",
        "mode": "bidirectional"
      },
      "io_queue": {
        "count": 4,
        "depth": 32
      },
      "nqn": "nqn.1992-01.example.com:string",
      "subsystem": {
        "name": "subsystem1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      },
      "tls": {
        "configured_psk": "NVMeTLSkey-
1:01:VRLbtnN9AQb2WXW3c9+wEf/DRLz0QuLdbYvEhwtdWwNf9LrZ:",
        "key_type": "configured"
      }
    }
  ],
  "subsystem": {
```

```
  "name": "subsystem1",
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"tls": {
  "configured_psk": "NVMeTLSkey-
1:01:VRLbtnN9AQb2WXW3c9+wEf/DRLz0QuLdbYvEhwtdWwNf9LrZ:",
  "key_type": "configured"
}
}
```

Response

Status: 201, Created

Name	Type	Description
num_records	integer	The number of records in the response.
records	array[nvme_subsystem_host]	

Example response

```
{
  "num_records": 1,
  "records": [
    {
      "dh_hmac_chap": {
        "controller_secret_key": "DHHC-
1:00:ia6zGodOr4SEG0Zzaw398rpY0wqipUWj4jWjUh4HWUz6aQ2n:",
        "group_size": "string",
        "hash_function": "string",
        "host_secret_key": "DHHC-
1:00:ia6zGodOr4SEG0Zzaw398rpY0wqipUWj4jWjUh4HWUz6aQ2n:",
        "mode": "bidirectional"
      },
      "io_queue": {
        "count": 4,
        "depth": 32
      },
      "nqn": "nqn.1992-01.example.com:string",
      "priority": "string",
      "records": [
        {
          "dh_hmac_chap": {
            "controller_secret_key": "DHHC-
1:00:ia6zGodOr4SEG0Zzaw398rpY0wqipUWj4jWjUh4HWUz6aQ2n:",
            "group_size": "string",
            "hash_function": "string",
            "host_secret_key": "DHHC-
1:00:ia6zGodOr4SEG0Zzaw398rpY0wqipUWj4jWjUh4HWUz6aQ2n:",
            "mode": "bidirectional"
          },
          "io_queue": {
            "count": 4,
            "depth": 32
          },
          "nqn": "nqn.1992-01.example.com:string",
          "subsystem": {
            "name": "subsystem1",
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
          },
          "tls": {
            "configured_psk": "NVMeTLSkey-
1:01:VRLbtnN9AQb2WXW3c9+wEf/DRLz0QuLdbYvEhwtdWwNf9LrZ:",
            "key_type": "configured"
          }
        }
      ]
    }
  ]
}
```

```

    }
  ],
  "subsystem": {
    "name": "subsystem1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "tls": {
    "configured_psk": "NVMeTLSkey-
1:01:VRLbtnN9AQb2WXW3c9+wEf/DRLz0QuLdbYvEhwtdWwNf9LrZ:",
    "key_type": "configured"
  }
}
]
}

```

Headers

Name	Description	Type
Location	Useful for tracking the resource location	string

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
262186	The "records" array and other host properties are mutually exclusive.
72089705	The NVMe subsystem host already exists for the NVMe subsystem.
72089716	The DH-HMAC-CHAP secret property is invalid. DH-HMAC-CHAP secrets must be in the format "DHHC-1:0X:<Base 64 encoded key and CRC>:", where X represents 0, 1, or 3 indicating no hash function, SHA-256, and SHA-512 respectively.
72089771	The NQN is invalid. A non-empty qualifier is required after the prefix. An example of a valid NQN is <i>nqn.1992-01.com.example:string</i> .
72089772	The NQN is invalid. Add the prefix ' <i>nqn</i> '. An example of a valid NQN is <i>nqn.1992-01.com.example:string</i> .

Error Code	Description
72089773	The NQN is invalid. The date field must be formatted <i>yyyy-mm</i> . An example of a valid NQN is <i>nqn.1992-01.com.example:string</i> .
72090001	The NVMe subsystem does not exist.
72090003	A host to be added to an NVMe subsystem is missing the "nqn" property.
72090036	An NVMe subsystem host NQN is duplicated in the input.
72090041	An element in the "records" array contains an invalid property.
72090042	The DH-HMAC-CHAP secret property is required when setting any other NVMe in-band authentication properties for a host.
72090151	NVMe/TCP-TLS is not supported for the effective version of the cluster.
72090202	A provided NVMe subsystem host TLS configured PSK is not valid.
72090204	A TLS configured PSK was not provided when adding an NVMe subsystem host with the configured key type.
72090205	An invalid combination for the TLS key type and configured PSK values was provided when adding an NVMe subsystem host. When key type is "none", no configured PSK is allowed. When key type is "configured", a configured PSK is required.

Also see the table of common errors in the [Response body](#) overview section of this documentation.

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

dh_hmac_chap

A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.

Name	Type	Description
controller_secret_key	string	The controller secret for NVMe in-band authentication. The value of this property is used by the NVMe host to authenticate the NVMe controller while establishing a connection. If unset, the controller is not authenticated. When supplied, the property <code>host_secret_key</code> must also be supplied. Optional in POST. This property is write-only. The <code>mode</code> property can be used to identify if a controller secret has been set for the host, but the controller secret value cannot be read. To change the value, the host must be deleted from the subsystem and re-added.
group_size	string	The Diffie-Hellman group size for NVMe in-band authentication. When property <code>host_secret_key</code> is provided, this property defaults to <code>2048_bit</code>. When supplied, the property <code>host_secret_key</code> must also be supplied. Optional in POST.

Name	Type	Description
hash_function	string	The hash function for NVMe in-band authentication. When property <code>host_secret_key</code> is provided, this property defaults to <code>sha_256</code> . When supplied, the property <code>host_secret_key</code> must also be supplied. Optional in POST.
host_secret_key	string	The host secret for NVMe in-band authentication. The value of this property is used by the NVMe controller to authenticate the NVMe host while establishing a connection. If unset, no authentication is performed by the host or controller. This property must be supplied if any other NVMe in-band authentication properties are supplied. Optional in POST. This property is write-only. The <code>mode</code> property can be used to identify if a host secret has been set for the host, but the host secret value cannot be read. To change the value, the host must be deleted from the subsystem and re-added.

Name	Type	Description
mode	string	The expected NVMe in-band authentication mode for the host. This property is an indication of which secrets are configured for the host. When set to: • none: The host has neither the host nor controller secret configured, and no authentication is performed. • unidirectional: The host has a host secret configured. The controller will authenticate the host. • bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.

io_queue

The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.

Name	Type	Description
count	integer	The number of I/O queue pairs. Absence of this property in GET implies property priority is set and platform and transport protocol specific values for I/O queue count is being used. Valid in GET only.
depth	integer	The I/O queue depth. Absence of this property in GET implies property priority is set and platform and transport protocol specific values for I/O queue depth is being used. Valid in GET only.

io_queue

The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.

Name	Type	Description
count	integer	The number of I/O queue pairs. The default value is inherited from the owning NVMe subsystem.
depth	integer	The I/O queue depth. The default value is inherited from the owning NVMe subsystem.

subsystem

The NVMe subsystem to which the NVMe host has been provisioned.

Name	Type	Description
name	string	The name of the NVMe subsystem.
uuid	string	The unique identifier of the NVMe subsystem.

tls

A container for the configuration for NVMe/TCP-TLS transport session for the host.

Name	Type	Description
configured_psk	string	A user supplied pre-shared key (PSK) value in PSK Interchange Format. Optional in POST. The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree. This property is only allowed when <code>key_type</code> is configured. If <code>configured_psk</code> is supplied and <code>key_type</code> is unset, <code>key_type</code> defaults to <code>configured</code>. This property is write-only. The <code>key_type</code> property can be used to identify if a configured PSK has been set for the host, but the PSK value cannot be read. To change the value, the host must be deleted from the subsystem and re-added.

Name	Type	Description
key_type	string	The method by which the TLS pre-shared key (PSK) is configured for the host. Optional in POST. The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree. Possible values: • <code>none</code> - TLS is not configured for the host connection. No value is allowed for property <code>configured_psk</code>. • <code>configured</code> - A user supplied PSK is configured for the NVMe/TCP-TLS transport connection between the host and the NVMe subsystem. A valid value for property <code>configured_psk</code> is required. This property defaults to <code>none</code> unless a value is supplied for <code>configured_psk</code> in which case it defaults to <code>configured</code>.

records

The NVMe host provisioned to access NVMe namespaces mapped to a subsystem.

Name	Type	Description
dh_hmac_chap	dh_hmac_chap	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.

Name	Type	Description
nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target. Not allowed in POST when the <code>records</code> property is used.
subsystem	subsystem	The NVMe subsystem to which the NVMe host has been provisioned.
tls	tls	A container for the configuration for NVMe/TCP-TLS transport session for the host.

nvme_subsystem_host

The NVMe host provisioned to access NVMe namespaces mapped to a subsystem.

Name	Type	Description
dh_hmac_chap	dh_hmac_chap	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.
nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target.
priority	string	The host priority setting allocates appropriate NVMe I/O queues (count and depth) for the host to submit I/O commands. Absence of this property in GET implies <code>io_queue</code> count and I/O queue depth are being used.
records	array[records]	An array of NVMe hosts specified to add multiple NVMe hosts to an NVMe subsystem in a single API call. Valid in POST only.

Name	Type	Description
subsystem	subsystem	The NVMe subsystem to which the NVMe host has been provisioned.
tls	tls	A container for the configuration for NVMe/TCP-TLS transport session for the host.

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

returned_error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

Delete an NVMe subsystem host

```
DELETE /protocols/nvme/subsystems/{subsystem.uuid}/hosts/{nqn}
```

Introduced In: 9.6

Deletes an NVMe subsystem host from an NVMe subsystem.

Related ONTAP commands

- `vserver nvme subsystem host remove`

Learn more

- [DOC /protocols/nvme/subsystems](#)

Parameters

Name	Type	In	Required	Description
subsystem.uuid	string	path	True	The unique identifier of the NVMe subsystem.
nqn	string	path	True	The NVMe qualified name (NQN) used to identify the NVMe subsystem host.

Response

Status: 200, Ok

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
72089771	The NQN is invalid. A non-empty qualifier is required after the prefix. An example of a valid NQN is <i>nqn.1992-01.com.example:string</i> .
72089772	The NQN is invalid. Add the prefix 'nqn'. An example of a valid NQN is <i>nqn.1992-01.com.example:string</i> .
72089773	The NQN is invalid. The date field must be formatted <i>yyyy-mm</i> . An example of a valid NQN is <i>nqn.1992-01.com.example:string</i> .
72090001	The NVMe subsystem does not exist.
72090004	The NVMe subsystem host does not exist.

Also see the table of common errors in the [Response body](#) overview section of this documentation.

Name	Type	Description
error	returned_error	

Example error

```
{
  "error": {
    "arguments": [
      {
        "code": "string",
        "message": "string"
      }
    ],
    "code": "4",
    "message": "entry doesn't exist",
    "target": "uuid"
  }
}
```

Definitions

See Definitions

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

returned_error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

Retrieve an NVMe subsystem host

GET /protocols/nvme/subsystems/{subsystem.uuid}/hosts/{nqn}

Introduced In: 9.6

Retrieves an NVMe subsystem host of an NVMe subsystem.

Related ONTAP commands

- `vserver nvme subsystem host show`

Learn more

- [DOC /protocols/nvme/subsystems](#)

Parameters

Name	Type	In	Required	Description
subsystem.uuid	string	path	True	The unique identifier of the NVMe subsystem.
nqn	string	path	True	The NVMe qualified name (NQN) used to identify the NVMe subsystem host.
fields	array[string]	query	False	Specify the fields to return.

Response

Status: 200, Ok

Name	Type	Description
_links	_links	
dh_hmac_chap	dh_hmac_chap	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.

Name	Type	Description
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.
nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target.
priority	string	The host priority setting allocates appropriate NVMe I/O queues (count and depth) for the host to submit I/O commands. Absence of this property in GET implies io_queue count and I/O queue depth are being used.
subsystem	subsystem	The NVMe subsystem to which the NVMe host has been provisioned.
tls	tls	A container for the configuration for NVMe/TCP-TLS transport session for the host.

Example response

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "dh_hmac_chap": {
    "group_size": "string",
    "hash_function": "string",
    "mode": "bidirectional"
  },
  "io_queue": {
    "count": 4,
    "depth": 32
  },
  "nqn": "nqn.1992-01.example.com:string",
  "priority": "string",
  "subsystem": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "subsystem1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "tls": {
    "key_type": "configured"
  }
}
```

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
72090001	The NVMe subsystem does not exist.
72090004	The NVMe subsystem host does not exist.

Error Code	Description
72090022	The NVMe subsystem host does not exist in the specified subsystem.

Also see the table of common errors in the [Response body](#) overview section of this documentation.

Name	Type	Description
error	returned_error	

Example error

```
{
  "error": {
    "arguments": [
      {
        "code": "string",
        "message": "string"
      }
    ],
    "code": "4",
    "message": "entry doesn't exist",
    "target": "uuid"
  }
}
```

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

dh_hmac_chap

A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.

Name	Type	Description
group_size	string	The Diffie-Hellman group size for NVMe in-band authentication. When property <code>host_secret_key</code> is provided, this property defaults to <code>2048_bit</code> . When supplied, the property <code>host_secret_key</code> must also be supplied. Optional in POST.
hash_function	string	The hash function for NVMe in-band authentication. When property <code>host_secret_key</code> is provided, this property defaults to <code>sha_256</code> . When supplied, the property <code>host_secret_key</code> must also be supplied. Optional in POST.

Name	Type	Description
mode	string	The expected NVMe in-band authentication mode for the host. This property is an indication of which secrets are configured for the host. When set to: • none: The host has neither the host nor controller secret configured, and no authentication is performed. • unidirectional: The host has a host secret configured. The controller will authenticate the host. • bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.

io_queue

The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.

Name	Type	Description
count	integer	The number of I/O queue pairs. Absence of this property in GET implies property priority is set and platform and transport protocol specific values for I/O queue count is being used. Valid in GET only.
depth	integer	The I/O queue depth. Absence of this property in GET implies property priority is set and platform and transport protocol specific values for I/O queue depth is being used. Valid in GET only.

io_queue

The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.

Name	Type	Description
count	integer	The number of I/O queue pairs. The default value is inherited from the owning NVMe subsystem.
depth	integer	The I/O queue depth. The default value is inherited from the owning NVMe subsystem.

subsystem

The NVMe subsystem to which the NVMe host has been provisioned.

Name	Type	Description
_links	_links	
name	string	The name of the NVMe subsystem.
uuid	string	The unique identifier of the NVMe subsystem.

tls

A container for the configuration for NVMe/TCP-TLS transport session for the host.

Name	Type	Description
key_type	string	The method by which the TLS pre-shared key (PSK) is configured for the host. Optional in POST. The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree. Possible values: • <code>none</code> - TLS is not configured for the host connection. No value is allowed for property <code>configured_psk</code>. • <code>configured</code> - A user supplied PSK is configured for the NVMe/TCP-TLS transport connection between the host and the NVMe subsystem. A valid value for property <code>configured_psk</code> is required. This property defaults to <code>none</code> unless a value is supplied for <code>configured_psk</code> in which case it defaults to <code>configured</code>.

records

The NVMe host provisioned to access NVMe namespaces mapped to a subsystem.

Name	Type	Description
_links	_links	
dh_hmac_chap	dh_hmac_chap	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.

Name	Type	Description
nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target. Not allowed in POST when the <code>records</code> property is used.
subsystem	subsystem	The NVMe subsystem to which the NVMe host has been provisioned.
tls	tls	A container for the configuration for NVMe/TCP-TLS transport session for the host.

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

returned_error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

Remove an NVMe subsystem

DELETE `/protocols/nvme/subsystems/{uuid}`

Introduced In: 9.6

Removes an NVMe subsystem.

Related ONTAP commands

- `vserver nvme subsystem delete`

Learn more

- [DOC /protocols/nvme/subsystems](#)

Parameters

Name	Type	In	Required	Description
uuid	string	path	True	The unique identifier of the NVMe subsystem.
allow_delete_while_mapped	boolean	query	False	Allows for the deletion of a mapped NVMe subsystem.
allow_delete_with_hosts	boolean	query	False	Allows for the deletion of an NVMe subsystem with NVMe hosts.

Response

Status: 200, Ok

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
72090001	The NVMe subsystem does not exist.
72090023	The NVMe subsystem contains one or more mapped namespaces. Use the <code>allow_delete_while_mapped</code> query parameter to delete an NVMe subsystem with mapped NVMe namespaces.

Error Code	Description
72090024	The NVMe subsystem contains one or more NVMe hosts. Use the <code>allow_delete_with_hosts</code> query parameter to delete an NVMe subsystem with NVMe hosts.

Also see the table of common errors in the [Response body](#) overview section of this documentation.

Name	Type	Description
error	returned_error	

Example error

```
{
  "error": {
    "arguments": [
      {
        "code": "string",
        "message": "string"
      }
    ],
    "code": "4",
    "message": "entry doesn't exist",
    "target": "uuid"
  }
}
```

Definitions

See Definitions

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

returned_error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

Retrieve an NVMe subsystem

GET /protocols/nvme/subsystems/{uuid}

Introduced In: 9.6

Retrieves an NVMe subsystem.

Expensive properties

There is an added computational cost to retrieving values for these properties. They are not included by default in GET results and must be explicitly requested using the `fields` query parameter. See [Requesting specific fields](#) to learn more.

- `subsystem_maps.*`

Related ONTAP commands

- `vserver nvme subsystem host show`
- `vserver nvme subsystem map show`
- `vserver nvme subsystem show`

Learn more

- [DOC /protocols/nvme/subsystems](#)

Parameters

Name	Type	In	Required	Description
uuid	string	path	True	The unique identifier of the NVMe subsystem.
fields	array[string]	query	False	Specify the fields to return.

Response

Status: 200, Ok

Name	Type	Description
_links	_links	
comment	string	A configurable comment for the NVMe subsystem. Optional in POST and PATCH.
delete_on_unmap	boolean	An option that causes the subsystem to be deleted when the last subsystem map associated with it is deleted. Optional in POST and PATCH. This property defaults to <i>false</i> when the subsystem is created.
hosts	array[hosts]	The NVMe hosts configured for access to the NVMe subsystem. Optional in POST.
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.
name	string	The name of the NVMe subsystem. Once created, an NVMe subsystem cannot be renamed. Required in POST.

Name	Type	Description
os_type	string	The host operating system of the NVMe subsystem's hosts. Required in POST.
serial_number	string	The serial number of the NVMe subsystem.
subsystem_maps	array[subsystem_maps]	The NVMe namespaces mapped to the NVMe subsystem. There is an added computational cost to retrieving property values for <code>subsystem_maps</code>. They are not populated for a GET request unless explicitly requested using the <code>fields</code> query parameter. See Requesting specific fields to learn more.
svm	svm	SVM, applies only to SVM-scoped objects.
target_nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target.
uuid	string	The unique identifier of the NVMe subsystem.

Name	Type	Description
vendor_uuids	array[string]	Vendor-specific identifiers (UUIDs) optionally assigned to an NVMe subsystem when the subsystem is created. The identifiers are used to enable vendor-specific NVMe protocol features. The identifiers are provided by a host application vendor and shared with NetApp prior to a joint product release. Creating an NVMe subsystem with an unknown or non-specific identifier will have no effect on the NVMe subsystem. Refer to the ONTAP SAN Administration Guide for a list of the supported vendor-specific identifiers. After a subsystem is created, the vendor-specific identifiers cannot be changed or removed. Optional in POST. • Introduced in: 9.9 • readCreate: 1

Example response

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "comment": "string",
  "hosts": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "dh_hmac_chap": {
        "group_size": "string",
        "hash_function": "string",
        "mode": "bidirectional"
      },
      "nqn": "nqn.1992-01.example.com:string",
      "priority": "string",
      "tls": {
        "key_type": "configured"
      }
    }
  ],
  "io_queue": {
    "default": {
      "count": 4,
      "depth": 16
    }
  },
  "name": "subsystem1",
  "os_type": "string",
  "serial_number": "wCVsgFMiuMhVAAAAAAB",
  "subsystem_maps": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "anagrpid": "00103050h",
      "namespace": {
```

```

    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "/vol/vol1/namespace1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "nsid": "00000001h"
}
],
"svm": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "svm1",
  "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
},
"target_nqn": "nqn.1992-01.example.com:string",
"uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",
"vendor_uuids": [
  "1447f0f4-42e5-0dfc-871a-dc9b3f92d8f8"
]
}

```

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
72090001	The NVMe subsystem does not exist.

Also see the table of common errors in the [Response body](#) overview section of this documentation.

Name	Type	Description
error	returned_error	

Example error

```
{
  "error": {
    "arguments": [
      {
        "code": "string",
        "message": "string"
      }
    ],
    "code": "4",
    "message": "entry doesn't exist",
    "target": "uuid"
  }
}
```

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

dh_hmac_chap

A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.

Name	Type	Description
group_size	string	The Diffie-Hellman group size for NVMe in-band authentication. When property <code>host_secret_key</code> is provided, this property defaults to <code>2048_bit</code> . When supplied, the property <code>host_secret_key</code> must also be supplied. Optional in POST.
hash_function	string	The hash function for NVMe in-band authentication. When property <code>host_secret_key</code> is provided, this property defaults to <code>sha_256</code> . When supplied, the property <code>host_secret_key</code> must also be supplied. Optional in POST.

Name	Type	Description
mode	string	The expected NVMe in-band authentication mode for the host. This property is an indication of which secrets are configured for the host. When set to: • none: The host has neither the host nor controller secret configured, and no authentication is performed. • unidirectional: The host has a host secret configured. The controller will authenticate the host. • bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.

tls

A container for the configuration for NVMe/TCP-TLS transport session for the host.

Name	Type	Description
key_type	string	The method by which the TLS pre-shared key (PSK) is configured for the host. Optional in POST. The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree. Possible values: • <code>none</code> - TLS is not configured for the host connection. No value is allowed for property <code>configured_psk</code>. • <code>configured</code> - A user supplied PSK is configured for the NVMe/TCP-TLS transport connection between the host and the NVMe subsystem. A valid value for property <code>configured_psk</code> is required. This property defaults to <code>none</code> unless a value is supplied for <code>configured_psk</code> in which case it defaults to <code>configured</code>.

hosts

Name	Type	Description
_links	_links	
dh_hmac_chap	dh_hmac_chap	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.
nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target.

Name	Type	Description
priority	string	The host priority setting allocates appropriate NVMe I/O queues (count and depth) for the host to submit I/O commands. Absence of this property in GET implies user configured values of I/O queue count and I/O queue depth are being used.
tls	tls	A container for the configuration for NVMe/TCP-TLS transport session for the host.

default

The default I/O queue parameters inherited by NVMe hosts in the NVMe subsystem.

Name	Type	Description
count	integer	The number of host I/O queue pairs.
depth	integer	The host I/O queue depth.

io_queue

The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.

Name	Type	Description
default	default	The default I/O queue parameters inherited by NVMe hosts in the NVMe subsystem.

namespace

An NVMe namespace mapped to the NVMe subsystem.

Name	Type	Description
_links	_links	
name	string	The name of the NVMe namespace.
uuid	string	The unique identifier of the NVMe namespace.

subsystem_maps

An NVMe namespace mapped to the NVMe subsystem.

Name	Type	Description
_links	_links	
anagrpId	string	The Asymmetric Namespace Access Group ID (ANAGRPID) of the NVMe namespace. The format for an ANAGRPI is 8 hexadecimal digits (zero-filled) followed by a lower case "h".
namespace	namespace	An NVMe namespace mapped to the NVMe subsystem.
nsid	string	The NVMe namespace identifier. This is an identifier used by an NVMe controller to provide access to the NVMe namespace. The format for an NVMe namespace identifier is 8 hexadecimal digits (zero-filled) followed by a lower case "h".

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
_links	_links	
name	string	The name of the SVM. This field cannot be specified in a PATCH method.
uuid	string	The unique identifier of the SVM. This field cannot be specified in a PATCH method.

error_arguments

Name	Type	Description
code	string	Argument code

Name	Type	Description
message	string	Message argument

returned_error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

Update an NVMe subsystem

PATCH /protocols/nvme/subsystems/{uuid}

Introduced In: 9.6

Updates an NVMe subsystem.

Related ONTAP commands

- `vserver nvme subsystem modify`

Learn more

- [DOC /protocols/nvme/subsystems](#)

Parameters

Name	Type	In	Required	Description
uuid	string	path	True	The unique identifier of the NVMe subsystem.

Request Body

Name	Type	Description
comment	string	A configurable comment for the NVMe subsystem. Optional in POST and PATCH.
delete_on_unmap	boolean	An option that causes the subsystem to be deleted when the last subsystem map associated with it is deleted. Optional in POST and PATCH. This property defaults to <i>false</i> when the subsystem is created.
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.
serial_number	string	The serial number of the NVMe subsystem.
subsystem_maps	array[subsystem_maps]	The NVMe namespaces mapped to the NVMe subsystem. There is an added computational cost to retrieving property values for <code>subsystem_maps</code>. They are not populated for a GET request unless explicitly requested using the <code>fields</code> query parameter. See Requesting specific fields to learn more.
target_nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target.
uuid	string	The unique identifier of the NVMe subsystem.

Example request

```
{
  "comment": "string",
  "io_queue": {
    "default": {
      "count": 4,
      "depth": 16
    }
  },
  "serial_number": "wCVsgFMiuMhVAAAAAAB",
  "subsystem_maps": [
    {
      "anagrpid": "00103050h",
      "namespace": {
        "name": "/vol/vol1/namespace1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      },
      "nsid": "00000001h"
    }
  ],
  "target_nqn": "nqn.1992-01.example.com:string",
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
```

Response

Status: 200, Ok

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
72090001	The NVMe subsystem does not exist.

Also see the table of common errors in the [Response body](#) overview section of this documentation.

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

dh_hmac_chap

A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.

Name	Type	Description
mode	string	The expected NVMe in-band authentication mode for the host. This property is an indication of which secrets are configured for the host. When set to: • none: The host has neither the host nor controller secret configured, and no authentication is performed. • unidirectional: The host has a host secret configured. The controller will authenticate the host. • bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.

tls

A container for the configuration for NVMe/TCP-TLS transport session for the host.

hosts

default

The default I/O queue parameters inherited by NVMe hosts in the NVMe subsystem.

Name	Type	Description
count	integer	The number of host I/O queue pairs.

Name	Type	Description
depth	integer	The host I/O queue depth.

io_queue

The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.

Name	Type	Description
default	default	The default I/O queue parameters inherited by NVMe hosts in the NVMe subsystem.

namespace

An NVMe namespace mapped to the NVMe subsystem.

Name	Type	Description
name	string	The name of the NVMe namespace.
uuid	string	The unique identifier of the NVMe namespace.

subsystem_maps

An NVMe namespace mapped to the NVMe subsystem.

Name	Type	Description
anagrpid	string	The Asymmetric Namespace Access Group ID (ANAGRPID) of the NVMe namespace. The format for an ANAGRPIP is 8 hexadecimal digits (zero-filled) followed by a lower case "h".
namespace	namespace	An NVMe namespace mapped to the NVMe subsystem.

Name	Type	Description
nsid	string	The NVMe namespace identifier. This is an identifier used by an NVMe controller to provide access to the NVMe namespace. The format for an NVMe namespace identifier is 8 hexadecimal digits (zero-filled) followed by a lower case "h".

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
name	string	The name of the SVM. This field cannot be specified in a PATCH method.
uuid	string	The unique identifier of the SVM. This field cannot be specified in a PATCH method.

nvme_subsystem

An NVMe subsystem maintains configuration state and namespace access control for a set of NVMe-connected hosts.

Name	Type	Description
comment	string	A configurable comment for the NVMe subsystem. Optional in POST and PATCH.
delete_on_unmap	boolean	An option that causes the subsystem to be deleted when the last subsystem map associated with it is deleted. Optional in POST and PATCH. This property defaults to <i>false</i> when the subsystem is created.
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.

Name	Type	Description
serial_number	string	The serial number of the NVMe subsystem.
subsystem_maps	array[subsystem_maps]	The NVMe namespaces mapped to the NVMe subsystem. There is an added computational cost to retrieving property values for <code>subsystem_maps</code>. They are not populated for a GET request unless explicitly requested using the <code>fields</code> query parameter. See Requesting specific fields to learn more.
target_nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target.
uuid	string	The unique identifier of the NVMe subsystem.

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

returned_error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

Copyright information

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.