# Manage NVMe subsystems

## REST API reference

NetApp
September 12, 2025

This PDF was generated from https://docs.netapp.com/us-en/ontap-restapi-991/protocols_nvme_subsystems_endpoint_overview.html on September 12, 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 . . . 12
  - Error . . . 14
  - Definitions . . . 15
- Create an NVMe subsystem . . . 20
  - Required properties . . . 20
  - Related ONTAP commands . . . 20
  - Learn more . . . 20
  - Parameters . . . 20
  - Request Body . . . 21
  - Response . . . 23
  - Error . . . 26
  - Definitions . . . 26
- Retrieve NVMe subsystem hosts . . . 31
  - Expensive properties . . . 31
  - Related ONTAP commands . . . 31
  - Learn more . . . 31
  - Parameters . . . 31
  - Response . . . 32
  - Error . . . 33
  - Definitions . . . 34
- Add NVMe subsystem hosts . . . 37
  - Required properties . . . 37
  - Related ONTAP commands . . . 37
  - Learn more . . . 37
  - Parameters . . . 37
  - Request Body . . . 38
  - Response . . . 39
  - Error . . . 40
  - Definitions . . . 41
- Delete an NVMe subsystem host . . . 44
  - Related ONTAP commands . . . 44
  - Learn more . . . 44
  - Parameters . . . 44

Response . . . 44
Error . . . 44
Definitions . . . 45
Retrieve an NVMe subsystem host . . . 46
Related ONTAP commands . . . 46
Learn more . . . 46
Parameters . . . 46
Response . . . 47
Error . . . 48
Definitions . . . 49
Remove an NVMe subsystem . . . 51
Related ONTAP commands . . . 52
Learn more . . . 52
Parameters . . . 52
Response . . . 52
Error . . . 52
Definitions . . . 53
Retrieve an NVMe subsystem . . . 54
Expensive properties . . . 54
Related ONTAP commands . . . 54
Learn more . . . 55
Parameters . . . 55
Response . . . 55
Error . . . 59
Definitions . . . 59
Update an NVMe subsystem . . . 62
Related ONTAP commands . . . 62
Learn more . . . 62
Parameters . . . 62
Request Body . . . 62
Response . . . 64
Error . . . 64
Definitions . . . 65

# 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-uPLAAAAAAAD",
"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": "svm1",
},
"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",
"io_queue": {
  "count": 4,
  "depth": 32
},
}
```

#### 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 |
|---|---|---|---|---|
| serial_number | string | query | False | Filter by serial_number |
| name | string | query | False | Filter by name |
| hosts.nqn | string | query | False | Filter by hosts.nqn |
| subsystem_maps.nsid | string | query | False | Filter by subsystem_maps.nsid |
| subsystem_maps.anagrpid | string | query | False | Filter by subsystem_maps.anagrpid |
| 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 |
| comment | string | query | False | Filter by comment |
| vendor_uuids | string | query | False | Filter by vendor_uuids<br>• Introduced in: 9.9 |
| delete_on_unmap | boolean | query | False | Filter by delete_on_unmap<br>• Introduced in: 9.7 |
| svm.uuid | string | query | False | Filter by svm.uuid |

| Name | Type | In | Required | Description |
|---|---|---|---|---|
| svm.name | string | query | False | Filter by svm.name |
| uuid | string | query | False | Filter by uuid |
| target_nqn | string | query | False | Filter by target_nqn |
| io_queue.default.count | integer | query | False | Filter by io_queue.default.count |
| io_queue.default.depth | integer | query | False | Filter by io_queue.default.depth |
| os_type | string | query | False | Filter by os_type |
| 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.<br>• 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.<br>• Default value: 1<br>• Max value: 120<br>• Min value: 0 |

| Name | Type | In | Required | Description |
|---|---|---|---|---|
| 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 | Number of records. |
| records | array[nvme_subsystem] | |

**Example response**


```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "comment": "string",
      "hosts": [
        {
          "nqn": "nqn.1992-01.example.com:string"
        }
      ],
      "io_queue": {
        "default": {
          "count": "4",
          "depth": "16"
        }
      },
      "name": "subsystem1",
      "os_type": "string",
      "serial_number": "wCVsgFMiuMhVAAAAAAAB",
      "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 | 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 | |

hosts

| Name | Type | Description |
|---|---|---|
| nqn | string | The NVMe qualified name (NQN) used to identify the NVMe storage target. |

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

svm

| Name | Type | Description |
|---|---|---|
| _links | _links | |
| name | string | The name of the SVM. |
| uuid | string | The unique identifier of the SVM. |

nvme_subsystem

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

| 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. This property defaults to *false* 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. |

| Name | Type | Description |
| --- | --- | --- |
| subsystem_maps | array[subsystem_maps] | The NVMe namespaces mapped to the NVMe subsystem.<br><br>There is an added cost to retrieving property values for `subsystem_maps`. They are not populated for either a collection GET or an instance GET unless explicitly requested using the `fields` query parameter. See Requesting specific fields to learn more. |
| svm | svm | |
| 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.<br><br>• Introduced in: 9.9<br>• readCreate: 1 |

error_arguments

| Name | Type | Description |
|---|---|---|
| code | string | Argument code |
| message | string | Message argument |

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. |

# 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 subsytem 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.<br>• Default value: |

## 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. This property defaults to *false* 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. |

| Name | Type | Description |
| --- | --- | --- |
| subsystem_maps | array[subsystem_maps] | The NVMe namespaces mapped to the NVMe subsystem.<br><br>There is an added cost to retrieving property values for `subsystem_maps`. They are not populated for either a collection GET or an instance GET unless explicitly requested using the `fields` query parameter. See Requesting specific fields to learn more. |
| svm | svm | |
| 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.<br><br>• Introduced in: 9.9<br>• readCreate: 1 |

**Example request**


```
{
  "comment": "string",
  "hosts": [
    {
      "nqn": "nqn.1992-01.example.com:string"
    }
  ],
  "io_queue": {
    "default": {
      "count": "4",
      "depth": "16"
    }
  },
  "name": "subsystem1",
  "os_type": "string",
  "serial_number": "wCVsgFMiuMhVAAAAAAAB",
  "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"
  ]
}
```


## Response

```
Status: 201, Created
```

| Name | Type | Description |
| --- | --- | --- |
| num_records | integer | Number of records. |
| records | array[nvme_subsystem] | |

**Example response**


```
{
  "records": [
    {
      "comment": "string",
      "hosts": [
        {
          "nqn": "nqn.1992-01.example.com:string"
        }
      ],
      "io_queue": {
        "default": {
          "count": "4",
          "depth": "16"
        }
      },
      "name": "subsystem1",
      "os_type": "string",
      "serial_number": "wCVsgFMiuMhVAAAAAAAB",
      "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"
      ]
    }
  ]
}
```

## Error

```
Status: Default
```

ONTAP Error Response Codes

| Error Code | Description |
|---|---|
| 2621462 | The supplied SVM does not exist. |
| 2621706 | The specified `svm.uuid` and `svm.name` do not refer to the same SVM. |
| 2621707 | The svm.uuid or svm.name must be provided. |
| 72089635 | Setting vendor-specific UUIDs on NVMe subsystems is not supported until the effective cluster version is 9.9 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. |
| 72089771 | The NQN is invalid. A non-empty qualifier is required after the prefix. An example of a valid NQN is *nqn.1992-01.com.example:string*. |
| 72089772 | The NQN is invalid. Add the prefix *'nqn'*. An example of a valid NQN is *nqn.1992-01.com.example:string*. |
| 72089773 | The NQN is invalid. The date field must be formatted *yyyy-mm*. An example of a valid NQN is *nqn.1992-01.com.example:string*. |
| 72090025 | The NVMe subsystem already exists for the SVM. |
| 72090029 | The NVMe service does not exist. |
| 72090030 | A partial success occured while adding multiple NVMe subsystem hosts to an NVMe subsystem. |
| 72090035 | Passing NVMe subsystem host NQNs on NVMe subsystem POST requires an effective cluster version of 9.7 or later. |
| 72090036 | The `hosts.nqn` NVMe subsystem property must contain unique values. |

## Definitions

**See Definitions**

href

| Name | Type | Description |
|---|---|---|
| href | string | |

_links

hosts

| Name | Type | Description |
|---|---|---|
| nqn | string | The NVMe qualified name (NQN) used to identify the NVMe storage target. |

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. |

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

svm

| Name | Type | Description |
|---|---|---|
| name | string | The name of the SVM. |
| uuid | string | The unique identifier of the SVM. |

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. This property defaults to *false* 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.<br><br>There is an added cost to retrieving property values for `subsystem_maps`. They are not populated for either a collection GET or an instance GET unless explicitly requested using the `fields` query parameter. See Requesting specific fields to learn more. |
| svm | svm | |
| 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.<br>• Introduced in: 9.9<br>• readCreate: 1 |

error_arguments

| Name | Type | Description |
|---|---|---|
| code | string | Argument code |
| message | string | Message argument |

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 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 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. |
| return_records | boolean | query | False | The default is true for GET calls. When set to false, only the number of records is returned.<br>• 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.<br>• Default value: 1<br>• Max value: 120<br>• 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 | Number of records. |
| records | array[nvme_subsystem_host] | |

**Example response**


```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "io_queue": {
        "count": "4",
        "depth": "32"
      },
      "nqn": "nqn.1992-01.example.com:string",
      "subsystem": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      }
    }
  ]
}
```


## Error

```
Status: Default
```

ONTAP Error Response Codes

| Error Code | Description |
|---|---|
| 72090001 | The NVMe subsystem does not exist. |

| Name | Type | Description |
| --- | --- | --- |
| error | 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 | |

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 | |
| uuid | string | The unique identifier of the NVMe subsystem. |

records

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

| Name | Type | Description |
|---|---|---|
| _links | _links | |
| 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. Not allowed in POST when the `records` property is used. |
| subsystem | subsystem | The NVMe subsystem to which the NVMe host has been provisioned. |

nvme_subsystem_host

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

| Name | Type | Description |
|---|---|---|
| _links | _links | |
| 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. Not allowed in POST when the `records` property is used. |
| subsystem | subsystem | The NVMe subsystem to which the NVMe host has been provisioned. |

error_arguments

| Name | Type | Description |
|---|---|---|
| code | string | Argument code |
| message | string | Message argument |

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.<br>• Default value: |

## Request Body

| 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. Not allowed in POST when the `records` property is 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. |

### Example request


```
{
  "io_queue": {
    "count": "4",
    "depth": "32"
  },
  "nqn": "nqn.1992-01.example.com:string",
  "records": [
    {
      "io_queue": {
        "count": "4",
        "depth": "32"
      },
      "nqn": "nqn.1992-01.example.com:string",
      "subsystem": {
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      }
    }
  ],
  "subsystem": {
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  }
}
```


## Response

```
Status: 201, Created
```

| Name | Type | Description |
|---|---|---|
| num_records | integer | Number of records. |
| records | array[nvme_subsystem_host] | |

**Example response**


```
{
  "records": [
    {
      "io_queue": {
        "count": "4",
        "depth": "32"
      },
      "nqn": "nqn.1992-01.example.com:string",
      "records": [
        {
          "io_queue": {
            "count": "4",
            "depth": "32"
          },
          "nqn": "nqn.1992-01.example.com:string",
          "subsystem": {
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
          }
        }
      ],
      "subsystem": {
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      }
    }
  ]
}
```


## Error

```
Status: Default
```

ONTAP Error Response Codes

| Error Code | Description |
|---|---|
| 72089705 | The NVMe subsystem host already exists for the NVMe subsystem. |
| 72089771 | The NQN is invalid. A non-empty qualifier is required after the prefix. An example of a valid NQN is *nqn.1992-01.com.example:string*. |
| 72089772 | The NQN is invalid. Add the prefix *'nqn'*. An example of a valid NQN is *nqn.1992-01.com.example:string*. |

| Error Code | Description |
| --- | --- |
| 72089773 | The NQN is invalid. The date field must be formatted *yyyy-mm*. An example of a valid NQN is *nqn.1992-01.com.example:string*. |
| 72090001 | The NVMe subsystem does not exist. |
| 72090002 | The POST request of hosts to an NVMe subsystem can only contain an 'nqn' property or 'records' property, but not both. |
| 72090003 | The elements in the records array for a POST of hosts to an NVMe subsystem must contain only the nqn property. |

## Definitions

**See Definitions**

href

| Name | Type | Description |
|---|---|---|
| href | string | |

_links

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 |
|---|---|---|
| uuid | string | The unique identifier of the NVMe subsystem. |

records

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

| 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. Not allowed in POST when the `records` property is used. |

| Name | Type | Description |
|---|---|---|
| subsystem | subsystem | The NVMe subsystem to which the NVMe host has been provisioned. |

nvme_subsystem_host

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

| 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. Not allowed in POST when the `records` property is 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. |

error_arguments

| Name | Type | Description |
|---|---|---|
| code | string | Argument code |
| message | string | Message argument |

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. |

# 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 *nqn.1992-01.com.example:string*. |
| 72089772 | The NQN is invalid. Add the prefix *'nqn'*. An example of a valid NQN is *nqn.1992-01.com.example:string*. |
| 72089773 | The NQN is invalid. The date field must be formatted *yyyy-mm*. An example of a valid NQN is *nqn.1992-01.com.example:string*. |
| 72090001 | The NVMe subsystem does not exist. |

| Name | Type | Description |
|---|---|---|
| error | 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 |

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. |

| Name | Type | In | Required | Description |
|---|---|---|---|---|
| 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 | |
| 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. Not allowed in POST when the `records` property is used. |
| subsystem | subsystem | The NVMe subsystem to which the NVMe host has been provisioned. |

**Example response**


```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "io_queue": {
    "count": "4",
    "depth": "32"
  },
  "nqn": "nqn.1992-01.example.com:string",
  "subsystem": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  }
}
```


## Error

```
Status: Default
```

ONTAP Error Response Codes

| Error Code | Description |
|---|---|
| 72090001 | The NVMe subsystem does not exist. |

| Name | Type | Description |
|---|---|---|
| error | 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 | |

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 | |
| uuid | string | The unique identifier of the NVMe subsystem. |

records

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

| Name | Type | Description |
|---|---|---|
| _links | _links | |

| 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. Not allowed in POST when the `records` property is used. |
| subsystem | subsystem | The NVMe subsystem to which the NVMe host has been provisioned. |

error_arguments

| Name | Type | Description |
|---|---|---|
| code | string | Argument code |
| message | string | Message argument |

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 `allow_delete_while_mapped` 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 `allow_delete_with_hosts` query parameter to delete an NVMe subsystem with NVMe hosts. |

| Name | Type | Description |
| --- | --- | --- |
| error | 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 |

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 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. This property defaults to *false* 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.<br><br>There is an added cost to retrieving property values for `subsystem_maps`. They are not populated for either a collection GET or an instance GET unless explicitly requested using the `fields` query parameter. See Requesting specific fields to learn more. |
| svm | svm | |
| 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.<br>• Introduced in: 9.9<br>• readCreate: 1 |

**Example response**


```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "comment": "string",
  "hosts": [
    {
      "nqn": "nqn.1992-01.example.com:string"
    }
  ],
  "io_queue": {
    "default": {
      "count": "4",
      "depth": "16"
    }
  },
  "name": "subsystem1",
  "os_type": "string",
  "serial_number": "wCVsgFMiuMhVAAAAAAAB",
  "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 | 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 | |

hosts

| Name | Type | Description |
|---|---|---|
| nqn | string | The NVMe qualified name (NQN) used to identify the NVMe storage target. |

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 | 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 |
|---|---|---|
| _links | _links | |
| anagrpid | string | The Asymmetric Namespace Access Group ID (ANAGRPID) of the NVMe namespace.<br><br>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. |
| nsid | string | The NVMe namespace identifier. This is an identifier used by an NVMe controller to provide access to the NVMe namespace.<br><br>The format for an NVMe namespace identifier is 8 hexadecimal digits (zero-filled) followed by a lower case "h". |

svm

| Name | Type | Description |
|---|---|---|
| _links | _links | |
| name | string | The name of the SVM. |
| uuid | string | The unique identifier of the SVM. |

error_arguments

| Name | Type | Description |
|---|---|---|
| code | string | Argument code |
| message | string | Message argument |

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. This property defaults to *false* 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.<br><br>There is an added cost to retrieving property values for `subsystem_maps`. They are not populated for either a collection GET or an instance GET unless explicitly requested using the `fields` query parameter. See Requesting specific fields to learn more. |
| svm | svm | |
| 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": "wCVsgFMiuMhVAAAAAAAB",
  "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"
}
```


## Response

```
Status: 200, Ok
```

## Error

```
Status: Default
```

ONTAP Error Response Codes

| Error Code | Description |
|---|---|
| 72090001 | The NVMe subsystem does not exist. |

## Definitions

**See Definitions**

href

| Name | Type | Description |
|---|---|---|
| href | string | |

_links

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

svm

| Name | Type | Description |
|---|---|---|
| name | string | The name of the SVM. |
| uuid | string | The unique identifier of the SVM. |

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. This property defaults to *false* when the subsystem is created. |

| 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. |
| serial_number | string | The serial number of the NVMe subsystem. |
| subsystem_maps | array[subsystem_maps] | The NVMe namespaces mapped to the NVMe subsystem.<br><br>There is an added cost to retrieving property values for `subsystem_maps`. They are not populated for either a collection GET or an instance GET unless explicitly requested using the `fields` query parameter. See Requesting specific fields to learn more. |
| svm | svm | |
| 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 |

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. |

## 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.