



## **NVMe**

### **ONTAP REST API reference**

NetApp  
July 16, 2025

# Table of Contents

NVMe .....	1
NVMe overview .....	1
Overview .....	1
View NVMe interfaces .....	2
Protocols NVMe interfaces endpoint overview .....	2
Retrieve NVMe interfaces .....	5
Retrieve an NVMe interface .....	16
Manage NVMe services .....	24
Protocols NVMe services endpoint overview .....	24
Retrieve NVMe services .....	29
Create an NVMe service .....	59
Delete an NVMe service .....	81
Retrieve an NVMe service .....	83
Update an NVMe service .....	102
Retrieve NVMe protocol historical performance metrics .....	120
Retrieve historical performance metrics for the NVMe protocol service of an SVM for a specific time ..	135
View NVMe subsystem controllers .....	145
Protocols NVMe subsystem-controllers endpoint overview .....	145
Retrieve NVMe subsystem controllers .....	152
Retrieve an NVMe subsystem controller .....	166
Manage NVMe subsystem maps .....	176
Protocols NVMe subsystem-maps endpoint overview .....	176
Retrieve NVMe subsystem maps .....	180
Create an NVMe subsystem map .....	188
Delete an NVMe subsystem map .....	200
Retrieve an NVMe subsystem map .....	207
Manage NVMe subsystems .....	215
Protocols NVMe subsystems endpoint overview .....	215
Retrieve NVMe subsystems .....	223
Create an NVMe subsystem .....	240
Retrieve NVMe subsystem hosts .....	259
Add NVMe subsystem hosts .....	271
Delete an NVMe subsystem host .....	287
Retrieve an NVMe subsystem host .....	290
Remove an NVMe subsystem .....	301
Retrieve an NVMe subsystem .....	304
Update an NVMe subsystem .....	317
Manage NVMe namespaces .....	332
Storage namespaces endpoint overview .....	332
Retrieve NVMe namespaces .....	341
Create an NVMe namespace .....	388
Retrieve historical performance metrics for an NVMe namespace .....	433
Retrieve historical performance metrics for an NVMe namespace for a specific time .....	442

Delete an NVMe namespace . . . . .	447
Retrieve an NVMe namespace . . . . .	451
Update an NVMe namespace . . . . .	489

# NVMe

## NVMe overview

### Overview

The Non-Volatile Memory Express (NVMe) API endpoints and objects provide for configuration, provisioning and management of the NVMe-related objects. NVMe over Fabrics (NVMe-oF) refers to the extensions and changes to the base NVMe command set to support NVMe commands over a fabric interconnect and from multiple hosts simultaneously. ONTAP implements elements of both NVMe and NVMe-oF. Throughout this documentation, NVMe is generally used to refer to both NVMe and NVMe-oF.

### Fibre Channel Logins

Fibre Channel logins represent connections, formed by Fibre Channel initiators, that have successfully logged in to ONTAP. This represents the Fibre Channel login on which higher-level protocols such as Fibre Channel Protocol (FCP) and Non-Volatile Memory Express over Fibre Channel (NVMe over FC) rely.

The Fibre Channel logins REST API provides information about active Fibre Channel logins.

### NVMe Interfaces

NVMe interfaces are network interfaces configured to support an NVMe over Fabrics protocol. The NVMe interfaces are Fibre Channel interfaces supporting an NVMe-oF data protocol. Regardless of the underlying physical and data protocol, NVMe interfaces are treated equally for the host-side application configuration. This endpoint provides a consolidated view of all NVMe interfaces for the purpose of configuring host-side applications.

The NVMe interfaces REST API provides NVMe-specific information about network interfaces configured to support an NVMe-oF protocol.

### Learn More

- *Fibre Channel Interfaces* found in the *networking* section. Fibre Channel interfaces are the logical endpoints for Fibre Channel network connections to an SVM.

### NVMe Services

A Non-Volatile Memory Express (NVMe) service defines the properties of the NVMe controller target for an SVM. There can be at most one NVMe service for a given SVM. An SVM's NVMe service must be created before NVMe host initiators can connect to the SVM.

The Non-Volatile Memory Express (NVMe) service REST API allows you to create, update, delete, and discover NVMe services for SVMs.

### NVMe Subsystem Controllers

Non-Volatile Memory Express (NVMe) subsystem controllers represent dynamic connections between hosts and a storage solution.

The NVMe subsystem controllers REST API provides information about connected hosts.

## NVMe Subsystem Maps

An NVMe subsystem map is an association of an NVMe namespace with an NVMe subsystem. When an NVMe namespace is mapped to an NVMe subsystem, the NVMe subsystem's hosts are granted access to the NVMe namespace. The relationship between an NVMe subsystem and an NVMe namespace is one subsystem to many namespaces.

The NVMe subsystem map REST API allows you to create, delete, and discover NVMe subsystem maps.

## NVMe Subsystems

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. It also allows you to add and remove NVMe hosts that can access the subsystem and associated namespaces.

## NVMe Namespaces

An NVMe namespace is a collection of addressable logical blocks presented to hosts connected to the storage virtual machine using the NVMe over Fabrics protocol.

The NVMe namespace REST API allows you to create, update, delete, and discover NVMe namespaces.

# View NVMe interfaces

## Protocols NVMe interfaces endpoint overview

### Overview

NVMe interfaces are network interfaces configured to support an NVMe over Fabrics (NVMe-oF) protocol. The NVMe interfaces are Fibre Channel (FC) interfaces supporting an NVMe-oF data protocol. Regardless of the underlying physical and data protocol, NVMe interfaces are treated equally for host-side application configuration. This endpoint provides a consolidated view of all NVMe interfaces for the purpose of configuring host-side applications.

The NVMe interfaces REST API provides NVMe-specific information about network interfaces configured to support an NVMe-oF protocol.

NVMe interfaces must be created using the protocol-specific endpoints for FC interfaces. See [POST /network/fc/interfaces](#) . After creation, the interfaces are available via this interface.

### Examples

#### Retrieving summary information for all NVMe interfaces

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

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

```
# The response:
{
  "records": [
    {
      "svm": {
        "uuid": "013e2c44-0d30-11e9-a684-005056bbdb14",
        "name": "svm1",
        "_links": {
          "self": {
            "href": "/api/svm/svms/013e2c44-0d30-11e9-a684-005056bbdb14"
          }
        }
      },
      "uuid": "74d69872-0d30-11e9-a684-005056bbdb14",
      "name": "nvme1",
      "_links": {
        "self": {
          "href": "/api/protocols/nvme/interfaces/74d69872-0d30-11e9-a684-005056bbdb14"
        }
      }
    },
    {
      "svm": {
        "uuid": "013e2c44-0d30-11e9-a684-005056bbdb14",
        "name": "svm1",
        "_links": {
          "self": {
            "href": "/api/svm/svms/013e2c44-0d30-11e9-a684-005056bbdb14"
          }
        }
      },
      "uuid": "77ded991-0d30-11e9-a684-005056bbdb14",
      "name": "nvme2",
      "_links": {
        "self": {
          "href": "/api/protocols/nvme/interfaces/77ded991-0d30-11e9-a684-005056bbdb14"
        }
      }
    }
  ],
  "num_records": 2,
  "_links": {
    "self": {

```

```
    "href": "/api/protocols/nvme/interfaces"
  }
}
```

### Retrieving detailed information for a specific NVMe interface

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

# The call:
curl -X GET 'https://<mgmt-ip>/api/protocols/nvme/interfaces/77ded991-0d30-11e9-a684-005056bbdb14' -H 'Accept: application/hal+json'

# The response:
{
  "svm": {
    "uuid": "013e2c44-0d30-11e9-a684-005056bbdb14",
    "name": "svm1",
    "_links": {
      "self": {
        "href": "/api/svm/svms/013e2c44-0d30-11e9-a684-005056bbdb14"
      }
    }
  },
  "uuid": "77ded991-0d30-11e9-a684-005056bbdb14",
  "name": "nvme2",
  "enabled": true,
  "node": {
    "name": "node1",
    "uuid": "cd4d47fd-0d2e-11e9-a684-005056bbdb14",
    "_links": {
      "self": {
        "href": "/api/cluster/nodes/cd4d47fd-0d2e-11e9-a684-005056bbdb14"
      }
    }
  },
  "transport_address": "nn-0x2003005056bbdb14:pn-0x2005005056bbdb14",
  "fc_interface": {
    "wwnn": "20:03:00:50:56:bb:db:14",
    "wwpn": "20:05:00:50:56:bb:db:14",
    "port": {
      "name": "1a",
      "uuid": "081ec491-0d2f-11e9-a684-005056bbdb14",
```

```

    "node": {
      "name": "node1"
    },
    "_links": {
      "self": {
        "href": "/api/network/fc/ports/081ec491-0d2f-11e9-a684-005056bbdb14"
      }
    },
    "_links": {
      "self": {
        "href": "/api/network/fc/interfaces/77ded991-0d30-11e9-a684-005056bbdb14"
      }
    },
    "_links": {
      "self": {
        "href": "/api/protocols/nvme/interfaces/77ded991-0d30-11e9-a684-005056bbdb14"
      }
    }
  }
}

```

## Retrieve NVMe interfaces

GET /protocols/nvme/interfaces

**Introduced In:** 9.6

Retrieves NVMe interfaces.

### Related ONTAP commands

- `vserver nvme show-interface`

### Learn more

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

### Parameters

Name	Type	In	Required	Description
svm.name	string	query	False	Filter by svm.name



Name	Type	In	Required	Description
svm.uuid	string	query	False	Filter by svm.uuid
name	string	query	False	Filter by name
transport_address	string	query	False	Filter by transport_address
uuid	string	query	False	Filter by uuid
transport_protocols	string	query	False	Filter by transport_protocols <ul style="list-style-type: none"> <li>• Introduced in: 9.10</li> </ul>
ip_interface.location.port.name	string	query	False	Filter by ip_interface.location.port.name <ul style="list-style-type: none"> <li>• Introduced in: 9.10</li> </ul>
ip_interface.location.port.node.name	string	query	False	Filter by ip_interface.location.port.node.name <ul style="list-style-type: none"> <li>• Introduced in: 9.10</li> </ul>
ip_interface.location.port.uuid	string	query	False	Filter by ip_interface.location.port.uuid <ul style="list-style-type: none"> <li>• Introduced in: 9.10</li> </ul>
ip_interface.ip.address	string	query	False	Filter by ip_interface.ip.address <ul style="list-style-type: none"> <li>• Introduced in: 9.10</li> </ul>
interface_type	string	query	False	Filter by interface_type <ul style="list-style-type: none"> <li>• Introduced in: 9.10</li> </ul>

Name	Type	In	Required	Description
fc_interface.wwpn	string	query	False	Filter by fc_interface.wwpn
fc_interface.wwnn	string	query	False	Filter by fc_interface.wwnn
fc_interface.port.node.name	string	query	False	Filter by fc_interface.port.node.name
fc_interface.port.name	string	query	False	Filter by fc_interface.port.name
fc_interface.port.uuid	string	query	False	Filter by fc_interface.port.uuid
enabled	boolean	query	False	Filter by enabled
node.uuid	string	query	False	Filter by node.uuid
node.name	string	query	False	Filter by node.name
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	<p>The default is true for GET calls. When set to false, only the number of records is returned.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> </ul>

Name	Type	In	Required	Description
return_timeout	integer	query	False	<p>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.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> <li>• Max value: 120</li> <li>• Min value: 0</li> </ul>
order_by	array[string]	query	False	Order results by specified fields and optional [asc

## Response

Status: 200, Ok

Name	Type	Description
_links	<a href="#">_links</a>	
num_records	integer	The number of records in the response.
records	array[ <a href="#">nvme_interface</a> ]	

## Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": 1,
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "fc_interface": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "port": {
          "_links": {
            "self": {
              "href": "/api/resourcelink"
            }
          },
          "name": "0a",
          "node": {
            "name": "node1"
          },
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
        },
        "wwnn": "20:00:00:50:56:b4:13:a9",
        "wwpn": "20:00:00:50:56:b4:13:a8"
      },
      "interface_type": "string",
      "ip_interface": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        }
      }
    }
  ]
}
```

```

    },
    "ip": {
      "address": "10.10.10.7"
    },
    "location": {
      "port": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "elb",
        "node": {
          "name": "node1"
        },
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      }
    },
    "name": "lif1",
    "node": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "node1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "svm": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "svm1",
      "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
    },
    "transport_address": "nn-0x200a00a0989062da:pn-0x200100a0989062da",
    "transport_protocols": [
      "string"
    ],
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  }
]

```

```
}
```

## Error

Status: Default, Error

Name	Type	Description
error	<a href="#">returned_error</a>	

## 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	<a href="#">href</a>	
self	<a href="#">href</a>	

\_links

Name	Type	Description
self	<a href="#">href</a>	

node

The node on which the FC port is located.

Name	Type	Description
name	string	The name of the node on which the FC port is located.

port

An FC port is the physical port of an FC adapter on a cluster node that can be connected to an FC network.

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	The name of the FC port.
node	<a href="#">node</a>	The node on which the FC port is located.
uuid	string	The unique identifier of the FC port.

fc\_interface

The attributes specific to a Fibre Channel-based NVMe interface.

This is populated when `interface_type` is *fc\_interface*.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
port	<a href="#">port</a>	An FC port is the physical port of an FC adapter on a cluster node that can be connected to an FC network.
wwnn	string	The WWNN (world wide node name) of the Fibre Channel NVMe interface.
wwpn	string	The WWPN (world wide port name) of the Fibre Channel NVMe interface.

self\_link

Name	Type	Description
self	<a href="#">href</a>	

ip

Name	Type	Description
address	string	IPv4 or IPv6 address

node

Name	Type	Description
name	string	Name of node on which the port is located.

port\_reference

Port UUID along with readable names. Either the UUID or both names may be supplied on input.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
name	string	
node	<a href="#">node</a>	
uuid	string	

location



Name	Type	Description
port	<a href="#">port_reference</a>	Port UUID along with readable names. Either the UUID or both names may be supplied on input.

ip\_interface

The attributes specific to an IP-based NVMe interface.

This is populated when `interface_type` is *ip\_interface*.

Name	Type	Description
_links	<a href="#">self_link</a>	
ip	<a href="#">ip</a>	
location	<a href="#">location</a>	

node

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	
uuid	string	

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
_links	<a href="#">_links</a>	
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\_interface

NVMe interfaces are network interfaces configured to support an NVMe over Fabrics (NVMe-oF) protocol. The NVMe interfaces are Fibre Channel interfaces supporting an NVMe-oF data protocol. Regardless of the underlying physical and data protocol, NVMe interfaces are treated equally for host-side application configuration. This endpoint provides a consolidated view of all NVMe interfaces for the purpose of configuring host-side applications.

NVMe interfaces must be created using the protocol-specific endpoints for Fibre Channel interfaces. See [POST /network/fc/interfaces](#) . After creation, the interfaces are available via this interface.

Name	Type	Description
_links	<a href="#">_links</a>	
enabled	boolean	The administrative state of the NVMe interface.
fc_interface	<a href="#">fc_interface</a>	The attributes specific to a Fibre Channel-based NVMe interface.  This is populated when <code>interface_type</code> is <code>fc_interface</code> .
interface_type	string	The underlying interface type of the NVMe interface. This property identifies which of <code>fc_interface</code> and <code>ip_interface</code> will be further populated.
ip_interface	<a href="#">ip_interface</a>	The attributes specific to an IP-based NVMe interface.  This is populated when <code>interface_type</code> is <code>ip_interface</code> .
name	string	The name of the NVMe interface.
node	<a href="#">node</a>	
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.
transport_address	string	The transport address of the NVMe interface.
transport_protocols	array[string]	The transport protocols supported by the NVMe interface.
uuid	string	The unique identifier of the NVMe interface.

error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

## Retrieve an NVMe interface

GET /protocols/nvme/interfaces/{uuid}

**Introduced In:** 9.6

Retrieves an NVMe interface.

### Related ONTAP commands

- `vserver nvme show-interface`

### Learn more

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

### Parameters

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

## Response

Status: 200, Ok

Name	Type	Description
_links	<a href="#">_links</a>	
enabled	boolean	The administrative state of the NVMe interface.
fc_interface	<a href="#">fc_interface</a>	<p>The attributes specific to a Fibre Channel-based NVMe interface.</p> <p>This is populated when <code>interface_type</code> is <i>fc_interface</i>.</p>
interface_type	string	The underlying interface type of the NVMe interface. This property identifies which of <i>fc_interface</i> and <i>ip_interface</i> will be further populated.
ip_interface	<a href="#">ip_interface</a>	<p>The attributes specific to an IP-based NVMe interface.</p> <p>This is populated when <code>interface_type</code> is <i>ip_interface</i>.</p>
name	string	The name of the NVMe interface.
node	<a href="#">node</a>	
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.
transport_address	string	The transport address of the NVMe interface.
transport_protocols	array[string]	The transport protocols supported by the NVMe interface.
uuid	string	The unique identifier of the NVMe interface.

## Example response

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "fc_interface": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "port": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "0a",
      "node": {
        "name": "node1"
      },
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "wwnn": "20:00:00:50:56:b4:13:a9",
    "wwpn": "20:00:00:50:56:b4:13:a8"
  },
  "interface_type": "string",
  "ip_interface": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "ip": {
      "address": "10.10.10.7"
    },
    "location": {
      "port": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        }
      }
    }
  },
}
```

```

    "name": "elb",
    "node": {
      "name": "node1"
    },
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  {
    "name": "lif1",
    "node": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "node1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "svm": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "svm1",
      "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
    },
    "transport_address": "nn-0x200a00a0989062da:pn-0x200100a0989062da",
    "transport_protocols": [
      "string"
    ],
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  }
}

```

## Error

Status: Default

## ONTAP Error Response Codes

Error Code	Description
2621462	The supplied SVM does not exist.

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

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	

node

The node on which the FC port is located.

Name	Type	Description
name	string	The name of the node on which the FC port is located.

port

An FC port is the physical port of an FC adapter on a cluster node that can be connected to an FC network.

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	The name of the FC port.
node	<a href="#">node</a>	The node on which the FC port is located.
uuid	string	The unique identifier of the FC port.

fc\_interface

The attributes specific to a Fibre Channel-based NVMe interface.

This is populated when `interface_type` is *fc\_interface*.

Name	Type	Description
_links	<a href="#">_links</a>	



Name	Type	Description
port	<a href="#">port</a>	An FC port is the physical port of an FC adapter on a cluster node that can be connected to an FC network.
wwnn	string	The WWNN (world wide node name) of the Fibre Channel NVMe interface.
wwpn	string	The WWPN (world wide port name) of the Fibre Channel NVMe interface.

self\_link

Name	Type	Description
self	<a href="#">href</a>	

ip

Name	Type	Description
address	string	IPv4 or IPv6 address

node

Name	Type	Description
name	string	Name of node on which the port is located.

port\_reference

Port UUID along with readable names. Either the UUID or both names may be supplied on input.

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	
node	<a href="#">node</a>	
uuid	string	

location

Name	Type	Description
port	<a href="#">port_reference</a>	Port UUID along with readable names. Either the UUID or both names may be supplied on input.

ip\_interface

The attributes specific to an IP-based NVMe interface.

This is populated when `interface_type` is *ip\_interface*.

Name	Type	Description
_links	<a href="#">self_link</a>	
ip	<a href="#">ip</a>	
location	<a href="#">location</a>	

node

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	
uuid	string	

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
_links	<a href="#">_links</a>	
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
message	string	Message argument

returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

## Manage NVMe services

### Protocols NVMe services endpoint overview

#### Overview

A Non-Volatile Memory Express (NVMe) service defines the properties of the NVMe controller target for an SVM. There can be at most one NVMe service for an SVM. An SVM's NVMe service must be created before NVMe host initiators can connect to the SVM.

The Non-Volatile Memory Express (NVMe) service REST API allows you to create, update, delete, and discover NVMe services for SVMs.

#### Performance monitoring

Performance of the SVM can be monitored by the `metric.*` and `statistics.*` properties. These show the performance of the SVM in terms of IOPS, latency and throughput. The `metric.*` properties denote an average whereas `statistics.*` properties denote a real-time monotonically increasing value aggregated across all nodes.

#### Examples

##### Creating an NVMe service for an SVM

The simplest way to create an NVMe service is to specify only the SVM, either by name or UUID. By default, the new NVMe service is enabled.

In this example, the `return_records` query parameter is used to retrieve the new NVMe service object in the REST response.

```

# The API:
POST /api/protocols/nvme/services

# The call:
curl -X POST 'https://<mgmt-
ip>/api/protocols/nvme/services?return_records=true' -H 'Accept:
application/hal+json' -d '{ "svm": { "name": "svm1" } }'

# The response:
{
  "num_records": 1,
  "records": [
    {
      "svm": {
        "uuid": "bfb1beb0-dc69-11e8-b29f-005056bb7341",
        "name": "svm1",
        "_links": {
          "self": {
            "href": "/api/svm/svms/bfb1beb0-dc69-11e8-b29f-005056bb7341"
          }
        }
      },
      "enabled": true,
      "_links": {
        "self": {
          "href": "/api/protocols/nvme/services/bfb1beb0-dc69-11e8-b29f-
005056bb7341"
        }
      }
    }
  ]
}

```

#### Retrieving the NVMe services for all SVMs in the cluster

```

# The API:
GET /api/protocols/nvme/services

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

# The response:

```

```

{
  "records": [
    {
      "svm": {
        "uuid": "ab60c350-dc68-11e8-9711-005056bbe408",
        "name": "svm0",
        "_links": {
          "self": {
            "href": "/api/svm/svms/ab60c350-dc68-11e8-9711-005056bbe408"
          }
        }
      },
      "_links": {
        "self": {
          "href": "/api/protocols/nvme/services/ab60c350-dc68-11e8-9711-005056bbe408"
        }
      }
    },
    {
      "svm": {
        "uuid": "bfb1beb0-dc69-11e8-b29f-005056bb7341",
        "name": "svm1",
        "_links": {
          "self": {
            "href": "/api/svm/svms/bfb1beb0-dc69-11e8-b29f-005056bb7341"
          }
        }
      },
      "_links": {
        "self": {
          "href": "/api/protocols/nvme/services/bfb1beb0-dc69-11e8-b29f-005056bb7341"
        }
      }
    }
  ],
  "num_records": 2,
  "_links": {
    "self": {
      "href": "/api/protocols/nvme/services"
    }
  }
}

```

### Retrieving details for a specific NVMe service

The NVMe service is identified by the UUID of its SVM.

```
# The API:
GET /api/protocols/nvme/services/{svm.uuid}

# The call:
curl -X GET 'https://<mgmt-ip>/api/protocols/nvme/services/bfb1beb0-dc69-11e8-b29f-005056bb7341' -H 'Accept: application/hal+json'

# The response:
{
  "svm": {
    "uuid": "bfb1beb0-dc69-11e8-b29f-005056bb7341",
    "name": "svm1",
    "_links": {
      "self": {
        "href": "/api/svm/svms/bfb1beb0-dc69-11e8-b29f-005056bb7341"
      }
    }
  },
  "enabled": true,
  "_links": {
    "self": {
      "href": "/api/protocols/nvme/services/bfb1beb0-dc69-11e8-b29f-005056bb7341"
    }
  }
}
```

### Disabling an NVMe service

Disabling an NVMe service shuts down all active NVMe connections for the SVM and prevents the creation of new NVMe connections.

The NVMe service to update is identified by the UUID of its SVM.

```
# The API:
PATCH /api/protocols/nvme/services/{svm.uuid}

# The call:
curl -X PATCH 'https://<mgmt-ip>/api/protocols/nvme/services/bfb1beb0-dc69-11e8-b29f-005056bb7341' -H 'Accept: application/hal+json' -d '{"enabled": "false" }'
```

You can retrieve the NVMe service to confirm the change.

```
# The API:
GET /api/protocols/nvme/services/{svm.uuid}

# The call:
curl -X GET 'https://<mgmt-ip>/api/protocols/nvme/services/bfb1beb0-dc69-11e8-b29f-005056bb7341' -H 'Accept: application/hal+json'

# The response:
{
  "svm": {
    "uuid": "bfb1beb0-dc69-11e8-b29f-005056bb7341",
    "name": "svm1",
    "_links": {
      "self": {
        "href": "/api/svm/svms/bfb1beb0-dc69-11e8-b29f-005056bb7341"
      }
    }
  },
  "enabled": false,
  "_links": {
    "self": {
      "href": "/api/protocols/nvme/services/bfb1beb0-dc69-11e8-b29f-005056bb7341"
    }
  }
}
```

### Deleting an NVMe service

The NVMe service must be disabled before it can be deleted. In addition, all NVMe interfaces, subsystems, and subsystem maps associated with the SVM must first be deleted.

The NVMe service to delete is identified by the UUID of its SVM.

```
# The API:
DELETE /api/protocols/nvme/services/{svm.uuid}

# The call:
curl -X DELETE 'https://<mgmt-ip>/api/protocols/nvme/services/bfblbeb0-
dc69-11e8-b29f-005056bb7341' -H 'Accept: application/hal+json'
```

## Retrieve NVMe services

GET /protocols/nvme/services

**Introduced In:** 9.6

Retrieves NVMe services.

### 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.

- `statistics.*`
- `metric.*`

### Related ONTAP commands

- `vserver nvme show`

### Learn more

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

### Parameters

Name	Type	In	Required	Description
statistics.latency_raw.write	integer	query	False	Filter by statistics.latency_raw.write <ul style="list-style-type: none"><li>• Introduced in: 9.7</li></ul>
statistics.latency_raw.total	integer	query	False	Filter by statistics.latency_raw.total <ul style="list-style-type: none"><li>• Introduced in: 9.7</li></ul>



Name	Type	In	Required	Description
statistics.latency_raw.read	integer	query	False	Filter by statistics.latency_raw.read  • Introduced in: 9.7
statistics.latency_raw.other	integer	query	False	Filter by statistics.latency_raw.other  • Introduced in: 9.7
statistics.status	string	query	False	Filter by statistics.status  • Introduced in: 9.7
statistics.timestamp	string	query	False	Filter by statistics.timestamp  • Introduced in: 9.7
statistics.fc.throughput_raw.total	integer	query	False	Filter by statistics.fc.throughput_raw.total  • Introduced in: 9.10
statistics.fc.throughput_raw.write	integer	query	False	Filter by statistics.fc.throughput_raw.write  • Introduced in: 9.10
statistics.fc.throughput_raw.read	integer	query	False	Filter by statistics.fc.throughput_raw.read  • Introduced in: 9.10

Name	Type	In	Required	Description
statistics.fc.iops_raw.write	integer	query	False	Filter by statistics.fc.iops_raw.write  • Introduced in: 9.10
statistics.fc.iops_raw.total	integer	query	False	Filter by statistics.fc.iops_raw.total  • Introduced in: 9.10
statistics.fc.iops_raw.read	integer	query	False	Filter by statistics.fc.iops_raw.read  • Introduced in: 9.10
statistics.fc.iops_raw.other	integer	query	False	Filter by statistics.fc.iops_raw.other  • Introduced in: 9.10
statistics.fc.latency_raw.write	integer	query	False	Filter by statistics.fc.latency_raw.write  • Introduced in: 9.10
statistics.fc.latency_raw.total	integer	query	False	Filter by statistics.fc.latency_raw.total  • Introduced in: 9.10
statistics.fc.latency_raw.read	integer	query	False	Filter by statistics.fc.latency_raw.read  • Introduced in: 9.10

Name	Type	In	Required	Description
statistics.fc.latency_raw.other	integer	query	False	Filter by statistics.fc.latency_raw.other  • Introduced in: 9.10
statistics.fc.status	string	query	False	Filter by statistics.fc.status  • Introduced in: 9.10
statistics.fc.timestamp	string	query	False	Filter by statistics.fc.timestamp  • Introduced in: 9.10
statistics.throughput_raw.total	integer	query	False	Filter by statistics.throughput_raw.total  • Introduced in: 9.7
statistics.throughput_raw.write	integer	query	False	Filter by statistics.throughput_raw.write  • Introduced in: 9.7
statistics.throughput_raw.read	integer	query	False	Filter by statistics.throughput_raw.read  • Introduced in: 9.7
statistics.tcp.throughput_raw.total	integer	query	False	Filter by statistics.tcp.throughput_raw.total  • Introduced in: 9.10

Name	Type	In	Required	Description
statistics.tcp.throughput_raw.write	integer	query	False	Filter by statistics.tcp.throughput_raw.write  • Introduced in: 9.10
statistics.tcp.throughput_raw.read	integer	query	False	Filter by statistics.tcp.throughput_raw.read  • Introduced in: 9.10
statistics.tcp.iops_raw.write	integer	query	False	Filter by statistics.tcp.iops_raw.write  • Introduced in: 9.10
statistics.tcp.iops_raw.total	integer	query	False	Filter by statistics.tcp.iops_raw.total  • Introduced in: 9.10
statistics.tcp.iops_raw.read	integer	query	False	Filter by statistics.tcp.iops_raw.read  • Introduced in: 9.10
statistics.tcp.iops_raw.other	integer	query	False	Filter by statistics.tcp.iops_raw.other  • Introduced in: 9.10
statistics.tcp.latency_raw.write	integer	query	False	Filter by statistics.tcp.latency_raw.write  • Introduced in: 9.10

Name	Type	In	Required	Description
statistics.tcp.latency_raw.total	integer	query	False	Filter by statistics.tcp.latency_raw.total  • Introduced in: 9.10
statistics.tcp.latency_raw.read	integer	query	False	Filter by statistics.tcp.latency_raw.read  • Introduced in: 9.10
statistics.tcp.latency_raw.other	integer	query	False	Filter by statistics.tcp.latency_raw.other  • Introduced in: 9.10
statistics.tcp.status	string	query	False	Filter by statistics.tcp.status  • Introduced in: 9.10
statistics.tcp.timestamp	string	query	False	Filter by statistics.tcp.timestamp  • Introduced in: 9.10
statistics.iops_raw.write	integer	query	False	Filter by statistics.iops_raw.write  • Introduced in: 9.7
statistics.iops_raw.total	integer	query	False	Filter by statistics.iops_raw.total  • Introduced in: 9.7

Name	Type	In	Required	Description
statistics.iops_raw.read	integer	query	False	Filter by statistics.iops_raw.read  • Introduced in: 9.7
statistics.iops_raw.other	integer	query	False	Filter by statistics.iops_raw.other  • Introduced in: 9.7
svm.name	string	query	False	Filter by svm.name
svm.uuid	string	query	False	Filter by svm.uuid
metric.fc.throughput.total	integer	query	False	Filter by metric.fc.throughput.total  • Introduced in: 9.10
metric.fc.throughput.write	integer	query	False	Filter by metric.fc.throughput.write  • Introduced in: 9.10
metric.fc.throughput.read	integer	query	False	Filter by metric.fc.throughput.read  • Introduced in: 9.10
metric.fc.duration	string	query	False	Filter by metric.fc.duration  • Introduced in: 9.10

Name	Type	In	Required	Description
metric.fc.latency.write	integer	query	False	Filter by metric.fc.latency.write  • Introduced in: 9.10
metric.fc.latency.total	integer	query	False	Filter by metric.fc.latency.total  • Introduced in: 9.10
metric.fc.latency.read	integer	query	False	Filter by metric.fc.latency.read  • Introduced in: 9.10
metric.fc.latency.other	integer	query	False	Filter by metric.fc.latency.other  • Introduced in: 9.10
metric.fc.timestamp	string	query	False	Filter by metric.fc.timestamp  • Introduced in: 9.10
metric.fc.iops.write	integer	query	False	Filter by metric.fc.iops.write  • Introduced in: 9.10
metric.fc.iops.total	integer	query	False	Filter by metric.fc.iops.total  • Introduced in: 9.10

Name	Type	In	Required	Description
metric.fc.iops.read	integer	query	False	Filter by metric.fc.iops.read  • Introduced in: 9.10
metric.fc.iops.other	integer	query	False	Filter by metric.fc.iops.other  • Introduced in: 9.10
metric.fc.status	string	query	False	Filter by metric.fc.status  • Introduced in: 9.10
metric.duration	string	query	False	Filter by metric.duration  • Introduced in: 9.7
metric.latency.write	integer	query	False	Filter by metric.latency.write  • Introduced in: 9.7
metric.latency.total	integer	query	False	Filter by metric.latency.total  • Introduced in: 9.7
metric.latency.read	integer	query	False	Filter by metric.latency.read  • Introduced in: 9.7
metric.latency.other	integer	query	False	Filter by metric.latency.other  • Introduced in: 9.7



Name	Type	In	Required	Description
metric.tcp.throughput.total	integer	query	False	Filter by metric.tcp.throughput.total  • Introduced in: 9.10
metric.tcp.throughput.write	integer	query	False	Filter by metric.tcp.throughput.write  • Introduced in: 9.10
metric.tcp.throughput.read	integer	query	False	Filter by metric.tcp.throughput.read  • Introduced in: 9.10
metric.tcp.duration	string	query	False	Filter by metric.tcp.duration  • Introduced in: 9.10
metric.tcp.latency.write	integer	query	False	Filter by metric.tcp.latency.write  • Introduced in: 9.10
metric.tcp.latency.total	integer	query	False	Filter by metric.tcp.latency.total  • Introduced in: 9.10
metric.tcp.latency.read	integer	query	False	Filter by metric.tcp.latency.read  • Introduced in: 9.10

Name	Type	In	Required	Description
metric.tcp.latency.other	integer	query	False	Filter by metric.tcp.latency.other  • Introduced in: 9.10
metric.tcp.timestamp	string	query	False	Filter by metric.tcp.timestamp  • Introduced in: 9.10
metric.tcp.iops.write	integer	query	False	Filter by metric.tcp.iops.write  • Introduced in: 9.10
metric.tcp.iops.total	integer	query	False	Filter by metric.tcp.iops.total  • Introduced in: 9.10
metric.tcp.iops.read	integer	query	False	Filter by metric.tcp.iops.read  • Introduced in: 9.10
metric.tcp.iops.other	integer	query	False	Filter by metric.tcp.iops.other  • Introduced in: 9.10
metric.tcp.status	string	query	False	Filter by metric.tcp.status  • Introduced in: 9.10
metric.iops.write	integer	query	False	Filter by metric.iops.write  • Introduced in: 9.7

Name	Type	In	Required	Description
metric.iops.total	integer	query	False	Filter by metric.iops.total  • Introduced in: 9.7
metric.iops.read	integer	query	False	Filter by metric.iops.read  • Introduced in: 9.7
metric.iops.other	integer	query	False	Filter by metric.iops.other  • Introduced in: 9.7
metric.timestamp	string	query	False	Filter by metric.timestamp  • Introduced in: 9.7
metric.status	string	query	False	Filter by metric.status  • Introduced in: 9.7
metric.throughput.total	integer	query	False	Filter by metric.throughput.total  • Introduced in: 9.7
metric.throughput.write	integer	query	False	Filter by metric.throughput.write  • Introduced in: 9.7

Name	Type	In	Required	Description
metric.throughput.read	integer	query	False	Filter by metric.throughput.read  • Introduced in: 9.7
enabled	boolean	query	False	Filter by enabled
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
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	<a href="#">_links</a>	
num_records	integer	The number of records in the response.
records	array[ <a href="#">nvme_service</a> ]	

## Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": 1,
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "metric": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        }
      },
      "duration": "PT15S",
      "fc": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        }
      },
      "duration": "PT15S",
      "iops": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "latency": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "status": "ok",
      "throughput": {
        "read": 200,
```

```

        "total": 1000,
        "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
},
{iops: {
    "read": 200,
    "total": 1000,
    "write": 100
},
latency: {
    "read": 200,
    "total": 1000,
    "write": 100
},
status: "ok",
tcp: {
    _links: {
        self: {
            href: "/api/resourcelink"
        }
    },
    duration: "PT15S",
    iops: {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    latency: {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    status: "ok",
    throughput: {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    timestamp: "2017-01-25 06:20:13 -0500"
},
throughput: {
    "read": 200,
    "total": 1000,
    "write": 100
},

```

```

    "timestamp": "2017-01-25 06:20:13 -0500"
  },
  "statistics": {
    "fc": {
      "iops_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "latency_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "status": "ok",
      "throughput_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "timestamp": "2017-01-25 06:20:13 -0500"
    },
    "iops_raw": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "latency_raw": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "status": "ok",
    "tcp": {
      "iops_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "latency_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "status": "ok",
      "throughput_raw": {

```



```

        "read": 200,
        "total": 1000,
        "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
},
"throughput_raw": {
    "read": 200,
    "total": 1000,
    "write": 100
},
"timestamp": "2017-01-25 06:20:13 -0500"
},
"svm": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
}
}
]
}

```

## Error

Status: Default, Error

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	
self	<a href="#">href</a>	

\_links

Name	Type	Description
self	<a href="#">href</a>	

iops

The rate of I/O operations observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

latency

The round trip latency in microseconds observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## throughput

The rate of throughput bytes per second observed at the storage object.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## fc

The NVMe/FC portion of the aggregated metrics.

Name	Type	Description
_links	<a href="#">_links</a>	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
iops	<a href="#">iops</a>	The rate of I/O operations observed at the storage object.

Name	Type	Description
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput	<a href="#">throughput</a>	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

tcp

The NVMe/TCP portion of the aggregated metrics.

Name	Type	Description
_links	<a href="#">_links</a>	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:

Name	Type	Description
iops	<a href="#">iops</a>	The rate of I/O operations observed at the storage object.
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput	<a href="#">throughput</a>	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

metric

Performance numbers, such as IOPS latency and throughput, for SVM protocols.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	

Name	Type	Description
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
fc	fc	The NVMe/FC portion of the aggregated metrics.
iops	iops	The rate of I/O operations observed at the storage object.
latency	latency	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
tcp	tcp	The NVMe/TCP portion of the aggregated metrics.
throughput	throughput	The rate of throughput bytes per second observed at the storage object.

Name	Type	Description
timestamp	string	The timestamp of the performance data.

#### iops\_raw

The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### latency\_raw

The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### throughput\_raw



Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

fc

The NVMe/FC portion of the aggregated statistics.

Name	Type	Description
iops_raw	<a href="#">iops_raw</a>	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.
latency_raw	<a href="#">latency_raw</a>	The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.

Name	Type	Description
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput_raw	throughput_raw	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.
timestamp	string	The timestamp of the performance data.

tcp

The NVMe/TCP portion of the aggregated statistics.

Name	Type	Description
iops_raw	iops_raw	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.

Name	Type	Description
latency_raw	<a href="#">latency_raw</a>	The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput_raw	<a href="#">throughput_raw</a>	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.
timestamp	string	The timestamp of the performance data.

#### statistics

These are raw performance numbers, such as IOPS latency and throughput for SVM protocols. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.

Name	Type	Description
fc	fc	The NVMe/FC portion of the aggregated statistics.
iops_raw	iops_raw	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.
latency_raw	latency_raw	The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
tcp	tcp	The NVMe/TCP portion of the aggregated statistics.

Name	Type	Description
throughput_raw	<a href="#">throughput_raw</a>	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.
timestamp	string	The timestamp of the performance data.

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
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\_service

A Non-Volatile Memory Express (NVMe) service defines the properties of the NVMe controller target for an SVM. There can be at most one NVMe service for an SVM. An SVM's NVMe service must be created before NVMe host initiators can connect to the SVM.

An NVMe service is identified by the UUID of its SVM.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
enabled	boolean	<p>The administrative state of the NVMe service. The NVMe service can be disabled to block all NVMe connectivity to the SVM.</p> <p>This is optional in POST and PATCH. The default setting is <i>true</i> (enabled) in POST.</p>
metric	<a href="#">metric</a>	Performance numbers, such as IOPS latency and throughput, for SVM protocols.

Name	Type	Description
statistics	<a href="#">statistics</a>	These are raw performance numbers, such as IOPS latency and throughput for SVM protocols. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.

#### error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

#### returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

## Create an NVMe service

POST /protocols/nvme/services

**Introduced In:** 9.6

Creates an NVMe service.

### Required properties

- `svm.uuid` or `svm.name` - The existing SVM in which to create the NVMe service.

## Related ONTAP commands

- `vserver nvme create`

## Learn more

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

## Parameters

Name	Type	In	Required	Description
return_records	boolean	query	False	<p>The default is false. If set to true, the records are returned.</p> <ul style="list-style-type: none"><li>• Default value:</li></ul>

## Request Body

Name	Type	Description
_links	<a href="#">_links</a>	
enabled	boolean	<p>The administrative state of the NVMe service. The NVMe service can be disabled to block all NVMe connectivity to the SVM.</p> <p>This is optional in POST and PATCH. The default setting is <i>true</i> (enabled) in POST.</p>
metric	<a href="#">metric</a>	Performance numbers, such as IOPS latency and throughput, for SVM protocols.
statistics	<a href="#">statistics</a>	These are raw performance numbers, such as IOPS latency and throughput for SVM protocols. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.

## Example request

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "metric": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    }
  },
  "duration": "PT15S",
  "fc": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    }
  },
  "duration": "PT15S",
  "iops": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "latency": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "status": "ok",
  "throughput": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "timestamp": "2017-01-25 06:20:13 -0500"
},
{iops": {
  "read": 200,
  "total": 1000,
  "write": 100
},
"latency": {
```



```

    "read": 200,
    "total": 1000,
    "write": 100
  },
  "status": "ok",
  "tcp": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "duration": "PT15S",
    "iops": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "latency": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "status": "ok",
    "throughput": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
  },
  "throughput": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "timestamp": "2017-01-25 06:20:13 -0500"
},
"statistics": {
  "fc": {
    "iops_raw": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "latency_raw": {
      "read": 200,

```

```

        "total": 1000,
        "write": 100
    },
    "status": "ok",
    "throughput_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
},
{iops_raw": {
    "read": 200,
    "total": 1000,
    "write": 100
},
"latency_raw": {
    "read": 200,
    "total": 1000,
    "write": 100
},
"status": "ok",
"tcp": {
    "iops_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "latency_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "status": "ok",
    "throughput_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
},
"throughput_raw": {
    "read": 200,
    "total": 1000,
    "write": 100
},

```

```

    "timestamp": "2017-01-25 06:20:13 -0500"
  },
  "svm": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  }
}

```

## Response

Status: 201, Created

Name	Type	Description
_links	<a href="#">_links</a>	
num_records	integer	The number of records in the response.
records	array[ <a href="#">nvme_service</a> ]	

## Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": 1,
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "metric": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        }
      },
      "duration": "PT15S",
      "fc": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        }
      },
      "duration": "PT15S",
      "iops": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "latency": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "status": "ok",
      "throughput": {
        "read": 200,
```

```

        "total": 1000,
        "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
},
"iops": {
    "read": 200,
    "total": 1000,
    "write": 100
},
"latency": {
    "read": 200,
    "total": 1000,
    "write": 100
},
"status": "ok",
"tcp": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "duration": "PT15S",
    "iops": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "latency": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "status": "ok",
    "throughput": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
},
"throughput": {
    "read": 200,
    "total": 1000,
    "write": 100
},

```

```

    "timestamp": "2017-01-25 06:20:13 -0500"
  },
  "statistics": {
    "fc": {
      "iops_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "latency_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "status": "ok",
      "throughput_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "timestamp": "2017-01-25 06:20:13 -0500"
    },
    "iops_raw": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "latency_raw": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "status": "ok",
    "tcp": {
      "iops_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "latency_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "status": "ok",
      "throughput_raw": {

```

```

        "read": 200,
        "total": 1000,
        "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
},
"throughput_raw": {
    "read": 200,
    "total": 1000,
    "write": 100
},
"timestamp": "2017-01-25 06:20:13 -0500"
},
"svm": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
}
}
]
}

```

## Headers

Name	Description	Type
Location	Useful for tracking the resource location	string

## Error

Status: Default

## ONTAP Error Response Codes

Error Code	Description
1115127	The cluster lacks a valid NVMe license.
2621462	The supplied SVM does not exist.
2621507	NVMe is not allowed for the specified SVM.

Error Code	Description
2621706	The specified <code>svm.uuid</code> and <code>svm.name</code> do not refer to the same SVM.
2621707	No SVM was specified. Either <code>svm.name</code> or <code>svm.uuid</code> must be supplied.
5374893	The SVM is stopped. The SVM must be running to create an NVMe service.
72089650	An NVMe service already exists for the specified SVM.
72089900	An NVMe service cannot be creating in an SVM that is configured for a SAN protocol.

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

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	

iops

The rate of I/O operations observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

latency

The round trip latency in microseconds observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.

Name	Type	Description
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## throughput

The rate of throughput bytes per second observed at the storage object.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## fc

The NVMe/FC portion of the aggregated metrics.

Name	Type	Description
_links	<a href="#">_links</a>	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
iops	<a href="#">iops</a>	The rate of I/O operations observed at the storage object.
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.

Name	Type	Description
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput	throughput	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

tcp

The NVMe/TCP portion of the aggregated metrics.

Name	Type	Description
_links	_links	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
iops	iops	The rate of I/O operations observed at the storage object.

Name	Type	Description
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput	<a href="#">throughput</a>	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

#### metric

Performance numbers, such as IOPS latency and throughput, for SVM protocols.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:

Name	Type	Description
fc	fc	The NVMe/FC portion of the aggregated metrics.
iops	iops	The rate of I/O operations observed at the storage object.
latency	latency	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
tcp	tcp	The NVMe/TCP portion of the aggregated metrics.
throughput	throughput	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

iops\_raw

The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### latency\_raw

The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### throughput\_raw

Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.

Name	Type	Description
read	integer	Performance metric for read I/O operations.

Name	Type	Description
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

fc

The NVMe/FC portion of the aggregated statistics.

Name	Type	Description
iops_raw	<a href="#">iops_raw</a>	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.
latency_raw	<a href="#">latency_raw</a>	The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.

Name	Type	Description
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput_raw	throughput_raw	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.
timestamp	string	The timestamp of the performance data.

tcp

The NVMe/TCP portion of the aggregated statistics.

Name	Type	Description
iops_raw	iops_raw	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.



Name	Type	Description
latency_raw	<a href="#">latency_raw</a>	The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput_raw	<a href="#">throughput_raw</a>	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.
timestamp	string	The timestamp of the performance data.

#### statistics

These are raw performance numbers, such as IOPS latency and throughput for SVM protocols. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.

Name	Type	Description
fc	fc	The NVMe/FC portion of the aggregated statistics.
iops_raw	iops_raw	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.
latency_raw	latency_raw	The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
tcp	tcp	The NVMe/TCP portion of the aggregated statistics.

Name	Type	Description
throughput_raw	<a href="#">throughput_raw</a>	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.
timestamp	string	The timestamp of the performance data.

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
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\_service

A Non-Volatile Memory Express (NVMe) service defines the properties of the NVMe controller target for an SVM. There can be at most one NVMe service for an SVM. An SVM's NVMe service must be created before NVMe host initiators can connect to the SVM.

An NVMe service is identified by the UUID of its SVM.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
enabled	boolean	<p>The administrative state of the NVMe service. The NVMe service can be disabled to block all NVMe connectivity to the SVM.</p> <p>This is optional in POST and PATCH. The default setting is <i>true</i> (enabled) in POST.</p>
metric	<a href="#">metric</a>	Performance numbers, such as IOPS latency and throughput, for SVM protocols.

Name	Type	Description
statistics	<a href="#">statistics</a>	These are raw performance numbers, such as IOPS latency and throughput for SVM protocols. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.

#### [\\_links](#)

Name	Type	Description
next	<a href="#">href</a>	
self	<a href="#">href</a>	

#### error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

#### returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

## Delete an NVMe service

DELETE /protocols/nvme/services/{svm.uuid}

**Introduced In:** 9.6

Deletes an NVMe service. An NVMe service must be disabled before it can be deleted. In addition, all NVMe

interfaces, subsystems, and subsystem maps associated with the SVM must first be deleted.

## Related ONTAP commands

- `vserver nvme delete`

## Learn more

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

## Parameters

Name	Type	In	Required	Description
svm.uuid	string	path	True	The unique identifier of the SVM whose NVMe service is to be deleted.

## Response

Status: 200, Ok

## Error

Status: Default

## ONTAP Error Response Codes

Error Code	Description
2621462	The supplied SVM does not exist.
72089651	The supplied SVM does not have an NVMe service.
72089653	There are subsystems associated with the NVMe service SVM. The subsystems must be removed before deleting the NVMe service.
72089654	There are NVMe-oF LIFs associated with the NVMe service SVM. The LIFs must be removed before deleting the NVMe service.
72090028	The NVMe service is enabled. The NVMe service must be disabled before it can be deleted.

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

Name	Type	Description
error	<a href="#">returned_error</a>	

## 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[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

## Retrieve an NVMe service

GET /protocols/nvme/services/{svm.uuid}

**Introduced In:** 9.6

Retrieves an NVMe service.

### Related ONTAP commands

- `vserver nvme show`

### Learn more

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

### Parameters

Name	Type	In	Required	Description
svm.uuid	string	path	True	The unique identifier of the SVM whose NVMe service is to be retrieved.
fields	array[string]	query	False	Specify the fields to return.

### Response

Status: 200, Ok

Name	Type	Description
_links	<a href="#">_links</a>	
enabled	boolean	<p>The administrative state of the NVMe service. The NVMe service can be disabled to block all NVMe connectivity to the SVM.</p> <p>This is optional in POST and PATCH. The default setting is <i>true</i> (enabled) in POST.</p>
metric	<a href="#">metric</a>	Performance numbers, such as IOPS latency and throughput, for SVM protocols.

Name	Type	Description
statistics	<a href="#">statistics</a>	These are raw performance numbers, such as IOPS latency and throughput for SVM protocols. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.



## Example response

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "metric": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    }
  },
  "duration": "PT15S",
  "fc": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    }
  },
  "duration": "PT15S",
  "iops": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "latency": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "status": "ok",
  "throughput": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "timestamp": "2017-01-25 06:20:13 -0500"
},
{iops": {
  "read": 200,
  "total": 1000,
  "write": 100
},
"latency": {
```

```

    "read": 200,
    "total": 1000,
    "write": 100
  },
  "status": "ok",
  "tcp": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "duration": "PT15S",
    "iops": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "latency": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "status": "ok",
    "throughput": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
  },
  "throughput": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "timestamp": "2017-01-25 06:20:13 -0500"
},
"statistics": {
  "fc": {
    "iops_raw": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "latency_raw": {
      "read": 200,

```

```

        "total": 1000,
        "write": 100
    },
    "status": "ok",
    "throughput_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
},
{iops_raw": {
    "read": 200,
    "total": 1000,
    "write": 100
},
"latency_raw": {
    "read": 200,
    "total": 1000,
    "write": 100
},
"status": "ok",
"tcp": {
    "iops_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "latency_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "status": "ok",
    "throughput_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
},
"throughput_raw": {
    "read": 200,
    "total": 1000,
    "write": 100
},

```

```

    "timestamp": "2017-01-25 06:20:13 -0500"
  },
  "svm": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  }
}

```

## Error

Status: Default

### ONTAP Error Response Codes

Error Code	Description
2621462	The supplied SVM does not exist.
72089651	The supplied SVM does not have an NVMe service.

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

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	

iops

The rate of I/O operations observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

latency

The round trip latency in microseconds observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.

Name	Type	Description
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## throughput

The rate of throughput bytes per second observed at the storage object.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## fc

The NVMe/FC portion of the aggregated metrics.

Name	Type	Description
_links	<a href="#">_links</a>	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
iops	<a href="#">iops</a>	The rate of I/O operations observed at the storage object.
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.

Name	Type	Description
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput	<a href="#">throughput</a>	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

## tcp

The NVMe/TCP portion of the aggregated metrics.

Name	Type	Description
_links	<a href="#">_links</a>	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
iops	<a href="#">iops</a>	The rate of I/O operations observed at the storage object.



Name	Type	Description
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput	<a href="#">throughput</a>	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

#### metric

Performance numbers, such as IOPS latency and throughput, for SVM protocols.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:

Name	Type	Description
fc	fc	The NVMe/FC portion of the aggregated metrics.
iops	iops	The rate of I/O operations observed at the storage object.
latency	latency	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
tcp	tcp	The NVMe/TCP portion of the aggregated metrics.
throughput	throughput	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

iops\_raw

The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### latency\_raw

The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### throughput\_raw

Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.

Name	Type	Description
read	integer	Performance metric for read I/O operations.

Name	Type	Description
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

fc

The NVMe/FC portion of the aggregated statistics.

Name	Type	Description
iops_raw	<a href="#">iops_raw</a>	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.
latency_raw	<a href="#">latency_raw</a>	The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.

Name	Type	Description
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput_raw	throughput_raw	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.
timestamp	string	The timestamp of the performance data.

tcp

The NVMe/TCP portion of the aggregated statistics.

Name	Type	Description
iops_raw	iops_raw	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.

Name	Type	Description
latency_raw	<a href="#">latency_raw</a>	The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput_raw	<a href="#">throughput_raw</a>	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.
timestamp	string	The timestamp of the performance data.

#### statistics

These are raw performance numbers, such as IOPS latency and throughput for SVM protocols. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.

Name	Type	Description
fc	fc	The NVMe/FC portion of the aggregated statistics.
iops_raw	iops_raw	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.
latency_raw	latency_raw	The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
tcp	tcp	The NVMe/TCP portion of the aggregated statistics.

Name	Type	Description
throughput_raw	<a href="#">throughput_raw</a>	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.
timestamp	string	The timestamp of the performance data.

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
_links	<a href="#">_links</a>	
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
message	string	Message argument

returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.



## Update an NVMe service

PATCH /protocols/nvme/services/{svm.uuid}

Introduced In: 9.6

Updates an NVMe service.

### Related ONTAP commands

- `vserver nvme modify`

### Learn more

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

### Parameters

Name	Type	In	Required	Description
svm.uuid	string	path	True	The unique identifier of the SVM whose NVMe service is to be updated.

### Request Body

Name	Type	Description
_links	<a href="#">_links</a>	
enabled	boolean	<p>The administrative state of the NVMe service. The NVMe service can be disabled to block all NVMe connectivity to the SVM.</p> <p>This is optional in POST and PATCH. The default setting is <i>true</i> (enabled) in POST.</p>
metric	<a href="#">metric</a>	Performance numbers, such as IOPS latency and throughput, for SVM protocols.
statistics	<a href="#">statistics</a>	These are raw performance numbers, such as IOPS latency and throughput for SVM protocols. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.

Name	Type	Description
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.

## Example request

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "metric": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    }
  },
  "duration": "PT15S",
  "fc": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    }
  },
  "duration": "PT15S",
  "iops": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "latency": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "status": "ok",
  "throughput": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "timestamp": "2017-01-25 06:20:13 -0500"
},
{iops": {
  "read": 200,
  "total": 1000,
  "write": 100
},
"latency": {
```

```

    "read": 200,
    "total": 1000,
    "write": 100
  },
  "status": "ok",
  "tcp": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "duration": "PT15S",
    "iops": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "latency": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "status": "ok",
    "throughput": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
  },
  "throughput": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "timestamp": "2017-01-25 06:20:13 -0500"
},
"statistics": {
  "fc": {
    "iops_raw": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "latency_raw": {
      "read": 200,

```

```

        "total": 1000,
        "write": 100
    },
    "status": "ok",
    "throughput_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
},
{iops_raw": {
    "read": 200,
    "total": 1000,
    "write": 100
},
"latency_raw": {
    "read": 200,
    "total": 1000,
    "write": 100
},
"status": "ok",
"tcp": {
    "iops_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "latency_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "status": "ok",
    "throughput_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
},
"throughput_raw": {
    "read": 200,
    "total": 1000,
    "write": 100
},

```

```

    "timestamp": "2017-01-25 06:20:13 -0500"
  },
  "svm": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  }
}

```

## Response

Status: 200, Ok

## Error

Status: Default

## ONTAP Error Response Codes

Error Code	Description
1115127	The cluster lacks a valid NVMe license.
2621462	The supplied SVM does not exist.
5374893	The SVM is stopped. The SVM must be running to create an NVMe service.
72089651	The supplied SVM does not have an NVMe service.

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

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	

iops

The rate of I/O operations observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

latency

The round trip latency in microseconds observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.



Name	Type	Description
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## throughput

The rate of throughput bytes per second observed at the storage object.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## fc

The NVMe/FC portion of the aggregated metrics.

Name	Type	Description
_links	<a href="#">_links</a>	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
iops	<a href="#">iops</a>	The rate of I/O operations observed at the storage object.
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.

Name	Type	Description
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput	throughput	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

tcp

The NVMe/TCP portion of the aggregated metrics.

Name	Type	Description
_links	_links	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
iops	iops	The rate of I/O operations observed at the storage object.

Name	Type	Description
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput	<a href="#">throughput</a>	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

#### metric

Performance numbers, such as IOPS latency and throughput, for SVM protocols.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:

Name	Type	Description
fc	fc	The NVMe/FC portion of the aggregated metrics.
iops	iops	The rate of I/O operations observed at the storage object.
latency	latency	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
tcp	tcp	The NVMe/TCP portion of the aggregated metrics.
throughput	throughput	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

iops\_raw

The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### latency\_raw

The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### throughput\_raw

Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.

Name	Type	Description
read	integer	Performance metric for read I/O operations.

Name	Type	Description
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

fc

The NVMe/FC portion of the aggregated statistics.

Name	Type	Description
iops_raw	<a href="#">iops_raw</a>	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.
latency_raw	<a href="#">latency_raw</a>	The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.

Name	Type	Description
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput_raw	throughput_raw	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.
timestamp	string	The timestamp of the performance data.

tcp

The NVMe/TCP portion of the aggregated statistics.

Name	Type	Description
iops_raw	iops_raw	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.

Name	Type	Description
latency_raw	<a href="#">latency_raw</a>	The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput_raw	<a href="#">throughput_raw</a>	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.
timestamp	string	The timestamp of the performance data.

#### statistics

These are raw performance numbers, such as IOPS latency and throughput for SVM protocols. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.



Name	Type	Description
fc	fc	The NVMe/FC portion of the aggregated statistics.
iops_raw	iops_raw	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.
latency_raw	latency_raw	The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
tcp	tcp	The NVMe/TCP portion of the aggregated statistics.

Name	Type	Description
throughput_raw	<a href="#">throughput_raw</a>	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.
timestamp	string	The timestamp of the performance data.

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
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\_service

A Non-Volatile Memory Express (NVMe) service defines the properties of the NVMe controller target for an SVM. There can be at most one NVMe service for an SVM. An SVM's NVMe service must be created before NVMe host initiators can connect to the SVM.

An NVMe service is identified by the UUID of its SVM.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
enabled	boolean	<p>The administrative state of the NVMe service. The NVMe service can be disabled to block all NVMe connectivity to the SVM.</p> <p>This is optional in POST and PATCH. The default setting is <i>true</i> (enabled) in POST.</p>
metric	<a href="#">metric</a>	Performance numbers, such as IOPS latency and throughput, for SVM protocols.

Name	Type	Description
statistics	<a href="#">statistics</a>	These are raw performance numbers, such as IOPS latency and throughput for SVM protocols. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.

#### error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

#### returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

## Retrieve NVMe protocol historical performance metrics

GET /protocols/nvme/services/{svm.uuid}/metrics

**Introduced In:** 9.7

Retrieves historical performance metrics for the NVMe protocol service of an SVM.

### Parameters

Name	Type	In	Required	Description
throughput.total	integer	query	False	Filter by throughput.total

Name	Type	In	Required	Description
throughput.write	integer	query	False	Filter by throughput.write
throughput.read	integer	query	False	Filter by throughput.read
timestamp	string	query	False	Filter by timestamp
iops.write	integer	query	False	Filter by iops.write
iops.total	integer	query	False	Filter by iops.total
iops.read	integer	query	False	Filter by iops.read
iops.other	integer	query	False	Filter by iops.other
status	string	query	False	Filter by status
duration	string	query	False	Filter by duration
tcp.iops.write	integer	query	False	Filter by tcp.iops.write  • Introduced in: 9.10
tcp.iops.total	integer	query	False	Filter by tcp.iops.total  • Introduced in: 9.10
tcp.iops.read	integer	query	False	Filter by tcp.iops.read  • Introduced in: 9.10
tcp.iops.other	integer	query	False	Filter by tcp.iops.other  • Introduced in: 9.10

Name	Type	In	Required	Description
tcp.status	string	query	False	Filter by tcp.status  • Introduced in: 9.10
tcp.throughput.total	integer	query	False	Filter by tcp.throughput.total  • Introduced in: 9.10
tcp.throughput.write	integer	query	False	Filter by tcp.throughput.write  • Introduced in: 9.10
tcp.throughput.read	integer	query	False	Filter by tcp.throughput.read  • Introduced in: 9.10
tcp.duration	string	query	False	Filter by tcp.duration  • Introduced in: 9.10
tcp.latency.write	integer	query	False	Filter by tcp.latency.write  • Introduced in: 9.10
tcp.latency.total	integer	query	False	Filter by tcp.latency.total  • Introduced in: 9.10
tcp.latency.read	integer	query	False	Filter by tcp.latency.read  • Introduced in: 9.10

Name	Type	In	Required	Description
tcp.latency.other	integer	query	False	Filter by tcp.latency.other  • Introduced in: 9.10
latency.write	integer	query	False	Filter by latency.write
latency.total	integer	query	False	Filter by latency.total
latency.read	integer	query	False	Filter by latency.read
latency.other	integer	query	False	Filter by latency.other
fc.iops.write	integer	query	False	Filter by fc.iops.write  • Introduced in: 9.10
fc.iops.total	integer	query	False	Filter by fc.iops.total  • Introduced in: 9.10
fc.iops.read	integer	query	False	Filter by fc.iops.read  • Introduced in: 9.10
fc.iops.other	integer	query	False	Filter by fc.iops.other  • Introduced in: 9.10
fc.status	string	query	False	Filter by fc.status  • Introduced in: 9.10
fc.throughput.total	integer	query	False	Filter by fc.throughput.total  • Introduced in: 9.10

Name	Type	In	Required	Description
fc.throughput.write	integer	query	False	Filter by fc.throughput.write  • Introduced in: 9.10
fc.throughput.read	integer	query	False	Filter by fc.throughput.read  • Introduced in: 9.10
fc.duration	string	query	False	Filter by fc.duration  • Introduced in: 9.10
fc.latency.write	integer	query	False	Filter by fc.latency.write  • Introduced in: 9.10
fc.latency.total	integer	query	False	Filter by fc.latency.total  • Introduced in: 9.10
fc.latency.read	integer	query	False	Filter by fc.latency.read  • Introduced in: 9.10
fc.latency.other	integer	query	False	Filter by fc.latency.other  • Introduced in: 9.10
svm.uuid	string	path	True	The unique identifier of the SVM.

Name	Type	In	Required	Description
interval	string	query	False	<p>The time range for the data. Examples can be 1h, 1d, 1m, 1w, 1y. The period for each time range is as follows:</p> <ul style="list-style-type: none"> <li>• 1h: Metrics over the most recent hour sampled over 15 seconds.</li> <li>• 1d: Metrics over the most recent day sampled over 5 minutes.</li> <li>• 1w: Metrics over the most recent week sampled over 30 minutes.</li> <li>• 1m: Metrics over the most recent month sampled over 2 hours.</li> <li>• 1y: Metrics over the most recent year sampled over a day.</li> <li>• Default value: 1</li> <li>• enum: ["1h", "1d", "1w", "1m", "1y"]</li> </ul>



Name	Type	In	Required	Description
return_timeout	integer	query	False	<p>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.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> <li>• Max value: 120</li> <li>• Min value: 0</li> </ul>
fields	array[string]	query	False	Specify the fields to return.
max_records	integer	query	False	Limit the number of records returned.
order_by	array[string]	query	False	Order results by specified fields and optional [asc
desc] direction. Default direction is 'asc' for ascending.	return_records	boolean	query	False

## Response

Status: 200, Ok

Name	Type	Description
_links	<a href="#">_links</a>	
num_records	integer	Number of records
records	array[ <a href="#">records</a> ]	

## Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": 1,
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "duration": "PT15S",
      "fc": {
        "duration": "PT15S",
        "iops": {
          "read": 200,
          "total": 1000,
          "write": 100
        },
        "latency": {
          "read": 200,
          "total": 1000,
          "write": 100
        },
        "status": "ok",
        "throughput": {
          "read": 200,
          "total": 1000,
          "write": 100
        }
      },
      "iops": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "latency": {
        "read": 200,
```

```

    "total": 1000,
    "write": 100
  },
  "status": "ok",
  "svm": {
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  },
  "tcp": {
    "duration": "PT15S",
    "iops": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "latency": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "status": "ok",
    "throughput": {
      "read": 200,
      "total": 1000,
      "write": 100
    }
  },
  "throughput": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "timestamp": "2017-01-25 06:20:13 -0500"
}
]
}

```

## Error

Status: Default

## ONTAP Error Response Codes

Error Code	Description
8585947	No metrics are available for the requested object.
8586225	An unexpected error occurred retrieving metrics for the requested object.

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

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	
self	<a href="#">href</a>	

\_links

Name	Type	Description
self	<a href="#">href</a>	

iops

The rate of I/O operations observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

latency

The round trip latency in microseconds observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## throughput

The rate of throughput bytes per second observed at the storage object.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## fc

The NVMe/FC portion of the aggregated metrics.

Name	Type	Description
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
iops	<a href="#">iops</a>	The rate of I/O operations observed at the storage object.

Name	Type	Description
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput	<a href="#">throughput</a>	The rate of throughput bytes per second observed at the storage object.

svm

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

tcp

The NVMe/TCP portion of the aggregated metrics.

Name	Type	Description
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
iops	<a href="#">iops</a>	The rate of I/O operations observed at the storage object.
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput	<a href="#">throughput</a>	The rate of throughput bytes per second observed at the storage object.

records

Performance numbers, such as IOPS latency and throughput, for SVM protocols.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	



Name	Type	Description
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
fc	fc	The NVMe/FC portion of the aggregated metrics.
iops	iops	The rate of I/O operations observed at the storage object.
latency	latency	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
svm	svm	
tcp	tcp	The NVMe/TCP portion of the aggregated metrics.

Name	Type	Description
throughput	<a href="#">throughput</a>	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

## Retrieve historical performance metrics for the NVMe protocol service of an SVM for a specific time

GET /protocols/nvme/services/{svm.uuid}/metrics/{timestamp}

**Introduced In:** 9.14

Retrieves historical performance metrics for the NVMe protocol service of an SVM for a specific time.

### Parameters

Name	Type	In	Required	Description
svm.uuid	string	path	True	The unique identifier of the SVM.

Name	Type	In	Required	Description
timestamp	string	path	True	The timestamp of the performance data.  • format: date-time
fields	array[string]	query	False	Specify the fields to return.

## Response

Status: 200, Ok

Name	Type	Description
_links	<a href="#">_links</a>	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
fc	<a href="#">fc</a>	The NVMe/FC portion of the aggregated metrics.
iops	<a href="#">iops</a>	The rate of I/O operations observed at the storage object.
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.

Name	Type	Description
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
svm	svm	
tcp	tcp	The NVMe/TCP portion of the aggregated metrics.
throughput	throughput	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

## Example response

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "duration": "PT15S",
  "fc": {
    "duration": "PT15S",
    "iops": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "latency": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "status": "ok",
    "throughput": {
      "read": 200,
      "total": 1000,
      "write": 100
    }
  },
  "iops": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "latency": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "status": "ok",
  "svm": {
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  },
  "tcp": {
    "duration": "PT15S",
    "iops": {
      "read": 200,
```

```

    "total": 1000,
    "write": 100
  },
  "latency": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "status": "ok",
  "throughput": {
    "read": 200,
    "total": 1000,
    "write": 100
  }
},
"throughput": {
  "read": 200,
  "total": 1000,
  "write": 100
},
"timestamp": "2017-01-25 06:20:13 -0500"
}

```

## Error

Status: Default

### ONTAP Error Response Codes

Error Code	Description
8585947	No metrics are available for the requested object.
8586225	An unexpected error occurred retrieving metrics for the requested object.

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

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	

iops

The rate of I/O operations observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

latency

The round trip latency in microseconds observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.



Name	Type	Description
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## throughput

The rate of throughput bytes per second observed at the storage object.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## fc

The NVMe/FC portion of the aggregated metrics.

Name	Type	Description
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
iops	<a href="#">iops</a>	The rate of I/O operations observed at the storage object.
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.

Name	Type	Description
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput	throughput	The rate of throughput bytes per second observed at the storage object.

#### svm

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

#### tcp

The NVMe/TCP portion of the aggregated metrics.

Name	Type	Description
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:

Name	Type	Description
iops	<a href="#">iops</a>	The rate of I/O operations observed at the storage object.
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput	<a href="#">throughput</a>	The rate of throughput bytes per second observed at the storage object.

#### error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

#### returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

## View NVMe subsystem controllers

### Protocols NVMe subsystem-controllers endpoint overview

#### Overview

Non-Volatile Memory Express (NVMe) subsystem controllers represent dynamic connections between hosts and a storage solution.

The NVMe subsystem controllers REST API provides information about connected hosts.

#### Examples

##### Retrieving the NVMe subsystem controllers for the entire system

```
# The API:
GET /api/protocols/nvme/subsystem-controllers

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

# The response:
{
  "records": [
    {
      "svm": {
        "uuid": "f0f5b928-2593-11e9-94c4-00a0989a1c8e",
        "name": "symmcon_fc_nvme_vserver_0",
        "_links": {
          "self": {
            "href": "/api/svm/svms/f0f5b928-2593-11e9-94c4-00a0989a1c8e"
          }
        }
      }
    },
  ],
}
```

```

    "subsystem": {
      "uuid": "14875240-2594-11e9-abde-00a098984313",
      "name": "symmcon_symmcon_fcnvme_vserver_0_subsystem_0",
      "_links": {
        "self": {
          "href": "/api/protocols/nvme/subsystems/14875240-2594-11e9-abde-00a098984313"
        }
      }
    },
    "id": "0040h",
    "_links": {
      "self": {
        "href": "/api/protocols/nvme/subsystem-controllers/14875240-2594-11e9-abde-00a098984313/0040h"
      }
    }
  },
  {
    "svm": {
      "uuid": "f0f5b928-2593-11e9-94c4-00a0989a1c8e",
      "name": "symmcon_fcnvme_vserver_0",
      "_links": {
        "self": {
          "href": "/api/svm/svms/f0f5b928-2593-11e9-94c4-00a0989a1c8e"
        }
      }
    },
    "subsystem": {
      "uuid": "14875240-2594-11e9-abde-00a098984313",
      "name": "symmcon_symmcon_fcnvme_vserver_0_subsystem_0",
      "_links": {
        "self": {
          "href": "/api/protocols/nvme/subsystems/14875240-2594-11e9-abde-00a098984313"
        }
      }
    },
    "id": "0041h",
    "_links": {
      "self": {
        "href": "/api/protocols/nvme/subsystem-controllers/14875240-2594-11e9-abde-00a098984313/0041h"
      }
    }
  }
},

```

```

{
  "svm": {
    "uuid": "f0f5b928-2593-11e9-94c4-00a0989a1c8e",
    "name": "symmcon_fcnvme_vserver_0",
    "_links": {
      "self": {
        "href": "/api/svm/svms/f0f5b928-2593-11e9-94c4-00a0989a1c8e"
      }
    }
  },
  "subsystem": {
    "uuid": "1489d0d5-2594-11e9-94c4-00a0989a1c8e",
    "name": "symmcon_symmcon_fcnvme_vserver_0_subsystem_1",
    "_links": {
      "self": {
        "href": "/api/protocols/nvme/subsystems/1489d0d5-2594-11e9-94c4-00a0989a1c8e"
      }
    }
  },
  "id": "0040h",
  "_links": {
    "self": {
      "href": "/api/protocols/nvme/subsystem-controllers/1489d0d5-2594-11e9-94c4-00a0989a1c8e/0040h"
    }
  }
},
{
  "svm": {
    "uuid": "f0f5b928-2593-11e9-94c4-00a0989a1c8e",
    "name": "symmcon_fcnvme_vserver_0",
    "_links": {
      "self": {
        "href": "/api/svm/svms/f0f5b928-2593-11e9-94c4-00a0989a1c8e"
      }
    }
  },
  "subsystem": {
    "uuid": "1489d0d5-2594-11e9-94c4-00a0989a1c8e",
    "name": "symmcon_symmcon_fcnvme_vserver_0_subsystem_1",
    "_links": {
      "self": {
        "href": "/api/protocols/nvme/subsystems/1489d0d5-2594-11e9-94c4-00a0989a1c8e"
      }
    }
  }
}

```

```

    }
  },
  "id": "0041h",
  "_links": {
    "self": {
      "href": "/api/protocols/nvme/subsystem-controllers/1489d0d5-2594-11e9-94c4-00a0989a1c8e/0041h"
    }
  }
},
"num_records": 4,
"_links": {
  "self": {
    "href": "/api/protocols/nvme/subsystem-controllers"
  }
}
}

```

#### Retrieving the NVMe subsystem controllers for a specific subsystem

```

# The API:
GET /api/protocols/nvme/subsystem-controllers

# The call:
curl -X GET 'https://<mgmt-ip>/api/protocols/nvme/subsystem-controllers?subsystem.uuid=14875240-2594-11e9-abde-00a098984313' -H 'Accept: application/hal+json'

# The response:
{
  "records": [
    {
      "svm": {
        "uuid": "f0f5b928-2593-11e9-94c4-00a0989a1c8e",
        "name": "symmcon_fc_nvme_vserver_0",
        "_links": {
          "self": {
            "href": "/api/svm/svms/f0f5b928-2593-11e9-94c4-00a0989a1c8e"
          }
        }
      },
      "subsystem": {
        "uuid": "14875240-2594-11e9-abde-00a098984313",

```

```

    "name": "symmcon_symmcon_fc_nvme_vserver_0_subsystem_0",
    "_links": {
      "self": {
        "href": "/api/protocols/nvme/subsystems/14875240-2594-11e9-abde-00a098984313"
      }
    },
    "id": "0040h",
    "_links": {
      "self": {
        "href": "/api/protocols/nvme/subsystem-controllers/14875240-2594-11e9-abde-00a098984313/0040h"
      }
    },
    {
      "svm": {
        "uuid": "f0f5b928-2593-11e9-94c4-00a0989a1c8e",
        "name": "symmcon_fc_nvme_vserver_0",
        "_links": {
          "self": {
            "href": "/api/svm/svms/f0f5b928-2593-11e9-94c4-00a0989a1c8e"
          }
        }
      },
      "subsystem": {
        "uuid": "14875240-2594-11e9-abde-00a098984313",
        "name": "symmcon_symmcon_fc_nvme_vserver_0_subsystem_0",
        "_links": {
          "self": {
            "href": "/api/protocols/nvme/subsystems/14875240-2594-11e9-abde-00a098984313"
          }
        }
      },
      "id": "0041h",
      "_links": {
        "self": {
          "href": "/api/protocols/nvme/subsystem-controllers/14875240-2594-11e9-abde-00a098984313/0041h"
        }
      }
    },
    "num_records": 2,

```



```
"_links": {
  "self": {
    "href": "/api/protocols/nvme/subsystem-controllers/14875240-2594-11e9-abde-00a098984313"
  }
}
}
```

### Retrieving a specific NVMe subsystem controller

```
# The API:
GET /api/protocols/nvme/subsystem-controllers/{subsystem.uuid}/{id}

# The call:
curl -X GET 'https://<mgmt-ip>/api/protocols/nvme/subsystem-controllers/14875240-2594-11e9-abde-00a098984313/0040h' -H 'Accept: application/hal+json'

# The response:
{
  "svm": {
    "uuid": "f0f5b928-2593-11e9-94c4-00a0989a1c8e",
    "name": "symmcon_fcnvme_vserver_0",
    "_links": {
      "self": {
        "href": "/api/svm/svms/f0f5b928-2593-11e9-94c4-00a0989a1c8e"
      }
    }
  },
  "subsystem": {
    "uuid": "14875240-2594-11e9-abde-00a098984313",
    "name": "symmcon_symmcon_fcnvme_vserver_0_subsystem_0",
    "_links": {
      "self": {
        "href": "/api/protocols/nvme/subsystems/14875240-2594-11e9-abde-00a098984313"
      }
    }
  },
  "id": "0040h",
  "interface": {
    "name": "symmcon_lif_fcnvme_symmcon_fcnvme_vserver_0_3a_0",
    "uuid": "fa1c5941-2593-11e9-94c4-00a0989a1c8e",
    "transport_address": "nn-0x200400a0989a1c8d:pn-0x200500a0989a1c8d",
```

```

    "_links": {
      "self": {
        "href": "/api/protocols/nvme/interfaces/falc5941-2593-11e9-94c4-00a0989a1c8e"
      }
    },
    "node": {
      "name": "ssan-8040-94a",
      "uuid": "ebf66f05-2590-11e9-abde-00a098984313",
      "_links": {
        "self": {
          "href": "/api/cluster/nodes/ebf66f05-2590-11e9-abde-00a098984313"
        }
      }
    },
    "host": {
      "transport_address": "nn-0x20000090fae00806:pn-0x10000090fae00806",
      "nqn": "nqn.2014-08.org.nvmexpress:uuid:c2846cb1-89d2-4020-a3b0-71ce907b4eef",
      "id": "b8546ca6097349e5b1558dc154fc073b"
    },
    "io_queue": {
      "count": 4,
      "depth": [
        32,
        32,
        32,
        32
      ]
    },
    "admin_queue": {
      "depth": 32
    },
    "dh_hmac_chap": {
      "mode": "none"
    },
    "keep_alive_timeout": 4000,
    "digest.header": true,
    "digest.data": false,
    "tls": {
      "key_type": "configured",
      "psk_identity": "NVMelR01 nqn.2014-08.org.nvmexpress:uuid:c2846cb1-89d2-4020-a3b0-71ce907b4eef nqn.1992-08.com.netapp:sn.ca3cae02070811ef9a53005056bb9001:subsystem.ss1c9X3RurQxGiGa76Tpk2tirifrUhHmVp035MOrtHXnAU=",

```

```
"cipher": "tls_aes_128_gcm_sha256"
},
"_links": {
  "self": {
    "href": "/api/protocols/nvme/subsystem-controllers/14875240-2594-11e9-abde-00a098984313/0040h"
  }
}
}
```

## Retrieve NVMe subsystem controllers

GET /protocols/nvme/subsystem-controllers

**Introduced In:** 9.6

Retrieves NVMe subsystem controllers.

### Related ONTAP commands

- `vserver nvme subsystem controller show`

### Learn more

- [DOC /protocols/nvme/subsystem-controllers](#)

### Parameters

Name	Type	In	Required	Description
svm.name	string	query	False	Filter by svm.name
svm.uuid	string	query	False	Filter by svm.uuid
admin_queue.depth	integer	query	False	Filter by admin_queue.depth
subsystem.name	string	query	False	Filter by subsystem.name <ul style="list-style-type: none"><li>• maxLength: 64</li><li>• minLength: 1</li></ul>
subsystem.uuid	string	query	False	Filter by subsystem.uuid
interface.uuid	string	query	False	Filter by interface.uuid

Name	Type	In	Required	Description
interface.name	string	query	False	Filter by interface.name
interface.transport_address	string	query	False	Filter by interface.transport_address
tls.key_type	string	query	False	Filter by tls.key_type <ul style="list-style-type: none"> <li>• Introduced in: 9.16</li> </ul>
tls.cipher	string	query	False	Filter by tls.cipher <ul style="list-style-type: none"> <li>• Introduced in: 9.16</li> </ul>
tls.psk_identity	string	query	False	Filter by tls.psk_identity <ul style="list-style-type: none"> <li>• Introduced in: 9.16</li> </ul>
io_queue.count	integer	query	False	Filter by io_queue.count
io_queue.depth	integer	query	False	Filter by io_queue.depth
digest.header	boolean	query	False	Filter by digest.header <ul style="list-style-type: none"> <li>• Introduced in: 9.15</li> </ul>
digest.data	boolean	query	False	Filter by digest.data <ul style="list-style-type: none"> <li>• Introduced in: 9.15</li> </ul>
dh_hmac_chap.mode	string	query	False	Filter by dh_hmac_chap.mode <ul style="list-style-type: none"> <li>• Introduced in: 9.12</li> </ul>

Name	Type	In	Required	Description
dh_hmac_chap.hash_function	string	query	False	Filter by dh_hmac_chap.hash_function  • Introduced in: 9.12
dh_hmac_chap.group_size	string	query	False	Filter by dh_hmac_chap.group_size  • Introduced in: 9.12
id	string	query	False	Filter by id
transport_protocol	string	query	False	Filter by transport_protocol  • Introduced in: 9.16
host.transport_address	string	query	False	Filter by host.transport_address
host.id	string	query	False	Filter by host.id
host.nqn	string	query	False	Filter by host.nqn  • maxLength: 223 • minLength: 1
keep_alive_timeout	integer	query	False	Filter by keep_alive_timeout  • Introduced in: 9.14
node.uuid	string	query	False	Filter by node.uuid
node.name	string	query	False	Filter by node.name
fields	array[string]	query	False	Specify the fields to return.

Name	Type	In	Required	Description
max_records	integer	query	False	Limit the number of records returned.
return_records	boolean	query	False	<p>The default is true for GET calls. When set to false, only the number of records is returned.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> </ul>
return_timeout	integer	query	False	<p>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.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> <li>• Max value: 120</li> <li>• Min value: 0</li> </ul>
order_by	array[string]	query	False	Order results by specified fields and optional [asc

## Response

Status: 200, Ok

Name	Type	Description
_links	<a href="#">_links</a>	
num_records	integer	The number of records in the response.
records	array[nvme_subsystem_controller]	

## Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": 1,
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "admin_queue": {
        "depth": 0
      },
      "dh_hmac_chap": {
        "group_size": "string",
        "hash_function": "string",
        "mode": "bidirectional"
      },
      "host": {
        "id": "b8546ca6097349e5b1558dc154fc073b",
        "nqn": "nqn.2014-08.org.nvmexpress:uuid:c2846cb1-89d2-4020-a3b0-71ce907b4eef",
        "transport_address": "nn-0x20000090fae00806:pn-0x10000090fae00806"
      },
      "id": "0040h",
      "interface": {
        "name": "lif1",
        "transport_address": "nn-0x200400a0989a1c8d:pn-0x200500a0989a1c8d",
        "uuid": "fa1c5941-2593-11e9-94c4-00a0989a1c8e"
      },
      "io_queue": {
        "count": 0,
        "depth": [
          "integer"
        ]
      }
    }
  ]
}
```

```

    },
    "keep_alive_timeout": 1500,
    "node": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "node1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "subsystem": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "subsystem1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "svm": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "svm1",
      "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
    },
    "tls": {
      "cipher": "tls_aes_128_gcm_sha256",
      "key_type": "configured",
      "psk_identity": "NVMelR01 nqn.2014-
08.org.nvmexpress:uuid:713b3816-f9bf-ba43-b95a-5e4bf8c726e9 nqn.1992-
08.com.netapp:sn.76f9d9bfb96511eea95e005056bb72b2:subsystem.ssl
mS1A7nrooevA9ZqAM09fQzWQlB2UZRt0BE1X4vINjY0="
    },
    "transport_protocol": "string"
  }
]
}

```

**Error**



Status: Default, Error

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	
self	<a href="#">href</a>	

\_links

Name	Type	Description
self	<a href="#">href</a>	

admin\_queue

Name	Type	Description
depth	integer	The depth of the admin queue for the controller.

dh\_hmac\_chap

A container for properties of the NVMe in-band authentication DH-HMAC-CHAP protocol used by the the host connection to the controller.

Name	Type	Description
group_size	string	The Diffie-Hellman group size used for NVMe in-band authentication. This property is populated only when NVMe in-band authentication was performed for the NVMe-oF transport connection.
hash_function	string	The hash function used for NVMe in-band authentication. This property is populated only when NVMe in-band authentication was performed for the NVMe-oF transport connection.

Name	Type	Description
mode	string	<p>The NVMe in-band authentication mode used for the host connection. When set to:</p> <ul style="list-style-type: none"> <li>• none: Neither the host nor controller was authenticated.</li> <li>• unidirectional: The controller authenticated the host.</li> <li>• bidirectional: The controller authenticated the host and the host authenticated the controller.</li> </ul>

## digest

Digests are properties of NVMe controllers created over the NVMe/TCP transport protocol. The usage of digests is negotiated between the host and the controller during connection setup. ONTAP enables digests only if the host requests them. The header digest is the crc32 checksum of the header portion of the NVMe/TCP PDU. The data digest is the crc32 checksum of the data portion of the NVMe/TCP PDU.

If a digest is enabled, upon receiving an NVMe/TCP PDU, ONTAP calculates the crc32 checksum of the associated portion of the PDU and compares it with the digest value present in the transmitted PDU. If there is a mismatch, ONTAP returns an error and destroys the controller.

Name	Type	Description
data	boolean	Reports if digests are enabled for the data portion of the PDU.
header	boolean	Reports if digests are enabled for the header portion of the PDU.

## host

Properties of the connected host.

Name	Type	Description
id	string	The host identifier registered with the controller.
nqn	string	The NVMe qualified name of the host.
transport_address	string	The transport address of the host.

## interface

The logical interface through which the host is connected.

Name	Type	Description
name	string	The name of the logical interface.
transport_address	string	The transport address of the logical interface.
uuid	string	The unique identifier of the logical interface.

io\_queue

Properties of the I/O queues available to the controller.

Name	Type	Description
count	integer	The number of I/O queues available to the controller.
depth	array[integer]	The depths of the I/O queues.

node

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	
uuid	string	

subsystem

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

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	The name of the NVMe subsystem.
uuid	string	The unique identifier of the NVMe subsystem.

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
_links	<a href="#">_links</a>	
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.

## tls

A container for properties that describe the encrypted NVMe/TCP transport connection between the host and the NVMe subsystem.

Name	Type	Description
cipher	string	The cipher suite used for the transport by the encrypted NVMe/TCP transport connection between the host and the NVMe subsystem. This property is populated only when encryption is in use for the transport connection.
key_type	string	<p>The method by which the TLS pre-shared key (PSK) was obtained when establishing the encrypted NVMe/TCP transport connection between the host and the NVMe subsystem. Possible values:</p> <ul style="list-style-type: none"> <li>• <code>none</code> - TLS encryption is not configured for the host connection.</li> <li>• <code>configured</code> - A user supplied PSK was used for the encrypted NVMe/TCP-TLS transport connection between the host and the NVMe subsystem.</li> </ul>

Name	Type	Description
psk_identity	string	The TLS PSK identity supplied by the host when establishing the encrypted NVMe/TCP transport connection between the host and the NVMe subsystem. This property is populated only when encryption is in use for the transport connection.

#### nvme\_subsystem\_controller

A Non-Volatile Memory Express (NVMe) subsystem controller represents a connection between a host and a storage solution.

An NVMe subsystem controller is identified by the NVMe subsystem UUID and the controller ID.

Name	Type	Description
_links	<a href="#">_links</a>	
admin_queue	<a href="#">admin_queue</a>	
dh_hmac_chap	<a href="#">dh_hmac_chap</a>	A container for properties of the NVMe in-band authentication DH-HMAC-CHAP protocol used by the the host connection to the controller.

Name	Type	Description
digest	<a href="#">digest</a>	<p>Digests are properties of NVMe controllers created over the NVMe/TCP transport protocol. The usage of digests is negotiated between the host and the controller during connection setup. ONTAP enables digests only if the host requests them. The header digest is the crc32 checksum of the header portion of the NVMe/TCP PDU. The data digest is the crc32 checksum of the data portion of the NVMe/TCP PDU.</p> <p>If a digest is enabled, upon receiving an NVMe/TCP PDU, ONTAP calculates the crc32 checksum of the associated portion of the PDU and compares it with the digest value present in the transmitted PDU. If there is a mismatch, ONTAP returns an error and destroys the controller.</p> <ul style="list-style-type: none"> <li>• readOnly: 1</li> <li>• Introduced in: 9.15</li> </ul>
host	<a href="#">host</a>	Properties of the connected host.
id	string	The identifier of the subsystem controller. This field consists of 4 zero-filled hexadecimal digits followed by an 'h'.
interface	<a href="#">interface</a>	The logical interface through which the host is connected.
io_queue	<a href="#">io_queue</a>	Properties of the I/O queues available to the controller.

Name	Type	Description
keep_alive_timeout	integer	<p>The keep-alive timeout value for the controller and all of its host connections, in milliseconds.</p> <p>If the NVMe controller does not receive a keep-alive request or an I/O request within the timeout window, the NVMe controller terminates its admin queue and I/O queue connections leading to NVMe controller teardown. If the NVMe host does not receive a response to a keep-alive request or an I/O request within the timeout window, the NVMe host initiates a connection disconnect.</p>
node	<a href="#">node</a>	
subsystem	<a href="#">subsystem</a>	An NVMe subsystem maintains configuration state and NVMe namespace access control for a set of NVMe-connected hosts.
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.
tls	<a href="#">tls</a>	A container for properties that describe the encrypted NVMe/TCP transport connection between the host and the NVMe subsystem.
transport_protocol	string	Transport Protocol

#### error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

#### returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments



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

## Retrieve an NVMe subsystem controller

GET /protocols/nvme/subsystem-controllers/{subsystem.uuid}/{id}

**Introduced In:** 9.6

Retrieves an NVMe subsystem controller.

### Related ONTAP commands

- `vserver nvme subsystem controller show`

### Learn more

- [DOC /protocols/nvme/subsystem-controllers](#)

### Parameters

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

### Response

Status: 200, Ok

Name	Type	Description
<code>_links</code>	<code>_links</code>	
<code>admin_queue</code>	<code>admin_queue</code>	
<code>dh_hmac_chap</code>	<code>dh_hmac_chap</code>	A container for properties of the NVMe in-band authentication DH-HMAC-CHAP protocol used by the the host connection to the controller.
<code>digest</code>	<code>digest</code>	<p>Digests are properties of NVMe controllers created over the NVMe/TCP transport protocol. The usage of digests is negotiated between the host and the controller during connection setup. ONTAP enables digests only if the host requests them. The header digest is the crc32 checksum of the header portion of the NVMe/TCP PDU. The data digest is the crc32 checksum of the data portion of the NVMe/TCP PDU.</p> <p>If a digest is enabled, upon receiving an NVMe/TCP PDU, ONTAP calculates the crc32 checksum of the associated portion of the PDU and compares it with the digest value present in the transmitted PDU. If there is a mismatch, ONTAP returns an error and destroys the controller.</p> <ul style="list-style-type: none"> <li>• readOnly: 1</li> <li>• Introduced in: 9.15</li> </ul>
<code>host</code>	<code>host</code>	Properties of the connected host.
<code>id</code>	<code>string</code>	The identifier of the subsystem controller. This field consists of 4 zero-filled hexadecimal digits followed by an 'h'.
<code>interface</code>	<code>interface</code>	The logical interface through which the host is connected.
<code>io_queue</code>	<code>io_queue</code>	Properties of the I/O queues available to the controller.

Name	Type	Description
keep_alive_timeout	integer	<p>The keep-alive timeout value for the controller and all of its host connections, in milliseconds.</p> <p>If the NVMe controller does not receive a keep-alive request or an I/O request within the timeout window, the NVMe controller terminates its admin queue and I/O queue connections leading to NVMe controller teardown. If the NVMe host does not receive a response to a keep-alive request or an I/O request within the timeout window, the NVMe host initiates a connection disconnect.</p>
node	node	
subsystem	subsystem	An NVMe subsystem maintains configuration state and NVMe namespace access control for a set of NVMe-connected hosts.
svm	svm	SVM, applies only to SVM-scoped objects.
tls	tls	A container for properties that describe the encrypted NVMe/TCP transport connection between the host and the NVMe subsystem.
transport_protocol	string	Transport Protocol

## Example response

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "admin_queue": {
    "depth": 0
  },
  "dh_hmac_chap": {
    "group_size": "string",
    "hash_function": "string",
    "mode": "bidirectional"
  },
  "host": {
    "id": "b8546ca6097349e5b1558dc154fc073b",
    "nqn": "nqn.2014-08.org.nvmexpress:uuid:c2846cb1-89d2-4020-a3b0-71ce907b4eef",
    "transport_address": "nn-0x20000090fae00806:pn-0x10000090fae00806"
  },
  "id": "0040h",
  "interface": {
    "name": "lif1",
    "transport_address": "nn-0x200400a0989a1c8d:pn-0x200500a0989a1c8d",
    "uuid": "fa1c5941-2593-11e9-94c4-00a0989a1c8e"
  },
  "io_queue": {
    "count": 0,
    "depth": [
      "integer"
    ]
  },
  "keep_alive_timeout": 1500,
  "node": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "node1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "subsystem": {
    "_links": {
```

```

    "self": {
      "href": "/api/resourcelink"
    },
    "name": "subsystem1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "svm": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  },
  "tls": {
    "cipher": "tls_aes_128_gcm_sha256",
    "key_type": "configured",
    "psk_identity": "NVMelR01 nqn.2014-08.org.nvmexpress:uuid:713b3816-
f9bf-ba43-b95a-5e4bf8c726e9 nqn.1992-
08.com.netapp:sn.76f9d9bfb96511eea95e005056bb72b2:subsystem.ss1
mS1A7nrooevA9ZqAM09fQzWQlB2UZRt0BE1X4vINjY0="
  },
  "transport_protocol": "string"
}

```

## Error

Status: Default

### ONTAP Error Response Codes

Error Code	Description
72090001	The supplied subsystem identifier does not exist.

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

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	

admin\_queue

Name	Type	Description
depth	integer	The depth of the admin queue for the controller.

dh\_hmac\_chap

A container for properties of the NVMe in-band authentication DH-HMAC-CHAP protocol used by the the host connection to the controller.

Name	Type	Description
group_size	string	The Diffie-Hellman group size used for NVMe in-band authentication. This property is populated only when NVMe in-band authentication was performed for the NVMe-oF transport connection.
hash_function	string	The hash function used for NVMe in-band authentication. This property is populated only when NVMe in-band authentication was performed for the NVMe-oF transport connection.

Name	Type	Description
mode	string	<p>The NVMe in-band authentication mode used for the host connection. When set to:</p> <ul style="list-style-type: none"> <li>• none: Neither the host nor controller was authenticated.</li> <li>• unidirectional: The controller authenticated the host.</li> <li>• bidirectional: The controller authenticated the host and the host authenticated the controller.</li> </ul>

## digest

Digests are properties of NVMe controllers created over the NVMe/TCP transport protocol. The usage of digests is negotiated between the host and the controller during connection setup. ONTAP enables digests only if the host requests them. The header digest is the crc32 checksum of the header portion of the NVMe/TCP PDU. The data digest is the crc32 checksum of the data portion of the NVMe/TCP PDU.

If a digest is enabled, upon receiving an NVMe/TCP PDU, ONTAP calculates the crc32 checksum of the associated portion of the PDU and compares it with the digest value present in the transmitted PDU. If there is a mismatch, ONTAP returns an error and destroys the controller.

Name	Type	Description
data	boolean	Reports if digests are enabled for the data portion of the PDU.
header	boolean	Reports if digests are enabled for the header portion of the PDU.

## host

Properties of the connected host.

Name	Type	Description
id	string	The host identifier registered with the controller.
nqn	string	The NVMe qualified name of the host.
transport_address	string	The transport address of the host.

## interface



The logical interface through which the host is connected.

Name	Type	Description
name	string	The name of the logical interface.
transport_address	string	The transport address of the logical interface.
uuid	string	The unique identifier of the logical interface.

io\_queue

Properties of the I/O queues available to the controller.

Name	Type	Description
count	integer	The number of I/O queues available to the controller.
depth	array[integer]	The depths of the I/O queues.

node

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	
uuid	string	

subsystem

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

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	The name of the NVMe subsystem.
uuid	string	The unique identifier of the NVMe subsystem.

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
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.

## tls

A container for properties that describe the encrypted NVMe/TCP transport connection between the host and the NVMe subsystem.

Name	Type	Description
cipher	string	The cipher suite used for the transport by the encrypted NVMe/TCP transport connection between the host and the NVMe subsystem. This property is populated only when encryption is in use for the transport connection.
key_type	string	<p>The method by which the TLS pre-shared key (PSK) was obtained when establishing the encrypted NVMe/TCP transport connection between the host and the NVMe subsystem. Possible values:</p> <ul style="list-style-type: none"> <li>• <code>none</code> - TLS encryption is not configured for the host connection.</li> <li>• <code>configured</code> - A user supplied PSK was used for the encrypted NVMe/TCP-TLS transport connection between the host and the NVMe subsystem.</li> </ul>

Name	Type	Description
psk_identity	string	The TLS PSK identity supplied by the host when establishing the encrypted NVMe/TCP transport connection between the host and the NVMe subsystem. This property is populated only when encryption is in use for the transport connection.

error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

## Manage NVMe subsystem maps

### Protocols NVMe subsystem-maps endpoint overview

#### Overview

An NVMe subsystem map is an association of an NVMe namespace with an NVMe subsystem. When an NVMe namespace is mapped to an NVMe subsystem, the NVMe subsystem's hosts are granted access to the NVMe namespace. The relationship between an NVMe subsystem and an NVMe namespace is one subsystem to many namespaces.

The NVMe subsystem map REST API allows you to create, delete and discover NVMe subsystem maps.

#### Examples

## Creating an NVMe subsystem map

```
# The API:
POST /api/protocols/nvme/subsystem-maps

# The call:
curl -X POST 'https://<mgmt-ip>/api/protocols/nvme/subsystem-maps' -H
'Accept: application/hal+json' -d '{ "svm": { "name": "svm1" },
"subsystem": { "name": "subsystem1" }, "namespace": { "name":
"/vol/vol1/namespace1" } }'
```

## Retrieving all of the NVMe subsystem maps

```
# The API:
GET /api/protocols/nvme/subsystem-maps

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

# The response:
{
  "records": [
    {
      "svm": {
        "uuid": "0e91b214-fe40-11e8-91a0-005056a79967",
        "name": "svm1",
        "_links": {
          "self": {
            "href": "/api/svm/svms/0e91b214-fe40-11e8-91a0-005056a79967"
          }
        }
      },
      "subsystem": {
        "uuid": "580a6b1e-fe43-11e8-91a0-005056a79967",
        "name": "subsystem1",
        "_links": {
          "self": {
            "href": "/api/protocols/nvme/subsystems/580a6b1e-fe43-11e8-91a0-005056a79967"
          }
        }
      },
      "namespace": {
```

```

    "uuid": "3ccdedc6-2519-4206-bc1f-b0f4adab6f89",
    "name": "/vol/vol1/namespace1",
    "_links": {
      "self": {
        "href": "/api/storage/namespaces/3ccdedc6-2519-4206-bc1f-
b0f4adab6f89"
      }
    },
    "_links": {
      "self": {
        "href": "/api/protocols/nvme/subsystem-maps/580a6b1e-fe43-11e8-
91a0-005056a79967/3ccdedc6-2519-4206-bc1f-b0f4adab6f89"
      }
    }
  ],
  "num_records": 1,
  "_links": {
    "self": {
      "href": "/api/protocols/nvme/subsystem-maps"
    }
  }
}

```

### Retrieving a specific NVMe subsystem map

The NVMe subsystem map is identified by the UUID of the NVMe subsystem followed by the UUID of the NVMe namespace.

```

# The API:
GET /api/protocols/nvme/subsystem-maps/{subsystem.uuid}/{namespace.uuid}

# The call:
curl -X GET 'https://<mgmt-ip>/api/protocols/nvme/subsystem-maps/580a6b1e-
fe43-11e8-91a0-005056a79967/3ccdedc6-2519-4206-bc1f-b0f4adab6f89' -H
'Accept: application/hal+json'

# The response:
{
  "svm": {
    "uuid": "0e91b214-fe40-11e8-91a0-005056a79967",
    "name": "svm1",
    "_links": {

```

```

    "self": {
      "href": "/api/svm/svms/0e91b214-fe40-11e8-91a0-005056a79967"
    }
  },
  "subsystem": {
    "uuid": "580a6b1e-fe43-11e8-91a0-005056a79967",
    "name": "subsystem1",
    "_links": {
      "self": {
        "href": "/api/protocols/nvme/subsystems/580a6b1e-fe43-11e8-91a0-005056a79967"
      }
    }
  },
  "namespace": {
    "uuid": "3ccdedc6-2519-4206-bc1f-b0f4adab6f89",
    "name": "/vol/vol1/namespace1",
    "node": {
      "name": "node1",
      "uuid": "012b4508-67d6-4788-8c2d-801f254ce976",
      "_links": {
        "self": {
          "href": "/api/cluster/nodes/012b4508-67d6-4788-8c2d-801f254ce976"
        }
      }
    }
  },
  "_links": {
    "self": {
      "href": "/api/storage/namespaces/3ccdedc6-2519-4206-bc1f-b0f4adab6f89"
    }
  },
  "nsid": "00000001h",
  "_links": {
    "self": {
      "href": "/api/protocols/nvme/subsystem-maps/580a6b1e-fe43-11e8-91a0-005056a79967/3ccdedc6-2519-4206-bc1f-b0f4adab6f89"
    }
  }
}

```

## Deleting an NVMe subsystem map

```
# The API:
DELETE /api/protocols/nvme/subsystem-
maps/{subsystem.uuid}/{namespace.uuid}

# The call:
curl -X DELETE 'https://<mgmt-ip>/api/protocols/nvme/subsystem-
maps/580a6b1e-fe43-11e8-91a0-005056a79967/3ccdedc6-2519-4206-bc1f-
b0f4adab6f89' -H 'Accept: application/hal+json'
```

## Retrieve NVMe subsystem maps

GET /protocols/nvme/subsystem-maps

**Introduced In:** 9.6

Retrieves NVMe subsystem maps.

### 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.

- `anagrpId`

### Related ONTAP commands

- `vserver nvme subsystem map show`

### Learn more

- [DOC /protocols/nvme/subsystem-maps](#)

### Parameters

Name	Type	In	Required	Description
svm.name	string	query	False	Filter by svm.name
svm.uuid	string	query	False	Filter by svm.uuid
subsystem.name	string	query	False	Filter by subsystem.name <ul style="list-style-type: none"><li>• maxLength: 64</li><li>• minLength: 1</li></ul>

Name	Type	In	Required	Description
subsystem.uuid	string	query	False	Filter by subsystem.uuid
nsid	string	query	False	Filter by nsid
namespace.uuid	string	query	False	Filter by namespace.uuid
namespace.node.uuid	string	query	False	Filter by namespace.node.uuid
namespace.node.name	string	query	False	Filter by namespace.node.name
namespace.name	string	query	False	Filter by namespace.name
anagrpId	string	query	False	Filter by anagrpId
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	<p>The default is true for GET calls. When set to false, only the number of records is returned.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> </ul>



Name	Type	In	Required	Description
return_timeout	integer	query	False	<p>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.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> <li>• Max value: 120</li> <li>• Min value: 0</li> </ul>
order_by	array[string]	query	False	Order results by specified fields and optional [asc

## Response

Status: 200, Ok

Name	Type	Description
_links	<a href="#">_links</a>	
num_records	integer	The number of records in the response.
records	array[nvme_subsystem_map]	

## Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": 1,
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "anagrp_id": "00103050h",
      "namespace": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "/vol/vol1/namespace1",
        "node": {
          "_links": {
            "self": {
              "href": "/api/resourcelink"
            }
          },
          "name": "node1",
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
        },
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      },
      "nsid": "00000001h",
      "subsystem": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "subsystem1",
```

```

    "uuid": "1cd8a442-86d1-11e0-a61c-123478563412"
  },
  "svm": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  }
}
]
}

```

## Error

Status: Default, Error

Name	Type	Description
error	<a href="#">returned_error</a>	

## 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	<a href="#">href</a>	
self	<a href="#">href</a>	

\_links

Name	Type	Description
self	<a href="#">href</a>	

node

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	
uuid	string	

namespace

The NVMe namespace to which the NVMe subsystem is mapped. Required in POST by supplying either the UUID, name, or both.

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	<p>The name of the NVMe namespace. Valid in POST. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.</p> <p>NVMe namespace names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;namespace&gt;"</code> where the qtree name is optional.</p>
node	<a href="#">node</a>	

Name	Type	Description
uuid	string	The unique identifier of the NVMe namespace. Valid in POST.

#### subsystem

The NVMe subsystem to which the NVMe namespace is mapped. Required in POST by supplying either `subsystem.uuid`, `subsystem.name` or both.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
name	string	The name of the NVMe subsystem.
uuid	string	The unique identifier of the NVMe subsystem.

#### svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
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\_map

An NVMe subsystem map is an association of an NVMe namespace with an NVMe subsystem. When an NVMe namespace is mapped to an NVMe subsystem, the NVMe subsystem's hosts are granted access to the NVMe namespace. The relationship between an NVMe subsystem and an NVMe namespace is one subsystem to many namespaces.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	

Name	Type	Description
anagrpId	string	<p>The Asymmetric Namespace Access Group ID (ANAGRPID) of the NVMe namespace.</p> <p>The format for an ANAGRPID is 8 hexadecimal digits (zero-filled) followed by a lower case "h".</p> <p>There is an added computational cost to retrieving this property's value. It is not populated for a GET request unless it is explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more.</p>
namespace	<a href="#">namespace</a>	The NVMe namespace to which the NVMe subsystem is mapped. Required in POST by supplying either the UUID, name, or both.
nsid	string	<p>The NVMe namespace identifier. This is an identifier used by an NVMe controller to provide access to the NVMe namespace.</p> <p>The format for an NVMe namespace identifier is 8 hexadecimal digits (zero-filled) followed by a lower case "h".</p>
subsystem	<a href="#">subsystem</a>	The NVMe subsystem to which the NVMe namespace is mapped. Required in POST by supplying either <code>subsystem.uuid</code> , <code>subsystem.name</code> or both.
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.

#### error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

## Create an NVMe subsystem map

POST /protocols/nvme/subsystem-maps

**Introduced In:** 9.6

Creates an NVMe subsystem map.

### Required properties

- `svm.uuid` or `svm.name` - Existing SVM in which to create the NVMe subsystem map.
- `namespace.uuid` or `namespace.name` - Existing NVMe namespace to map to the specified NVme subsystem.
- `subsystem.uuid` or `subsystem.name` - Existing NVMe subsystem to map to the specified NVMe namespace.

### Related ONTAP commands

- `vserver nvme subsystem map add`

### Learn more

- [DOC /protocols/nvme/subsystem-maps](#)

### Parameters

Name	Type	In	Required	Description
return_records	boolean	query	False	The default is false. If set to true, the records are returned. <ul style="list-style-type: none"><li>• Default value:</li></ul>

## Request Body

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
anagrpId	string	<p>The Asymmetric Namespace Access Group ID (ANAGRPID) of the NVMe namespace.</p> <p>The format for an ANAGRPID is 8 hexadecimal digits (zero-filled) followed by a lower case "h".</p> <p>There is an added computational cost to retrieving this property's value. It is not populated for a GET request unless it is explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more.</p>
namespace	<a href="#">namespace</a>	<p>The NVMe namespace to which the NVMe subsystem is mapped. Required in POST by supplying either the UUID, name, or both.</p>
nsid	string	<p>The NVMe namespace identifier. This is an identifier used by an NVMe controller to provide access to the NVMe namespace.</p> <p>The format for an NVMe namespace identifier is 8 hexadecimal digits (zero-filled) followed by a lower case "h".</p>
subsystem	<a href="#">subsystem</a>	<p>The NVMe subsystem to which the NVMe namespace is mapped. Required in POST by supplying either <code>subsystem.uuid</code>, <code>subsystem.name</code> or both.</p>
svm	<a href="#">svm</a>	<p>SVM, applies only to SVM-scoped objects.</p>



**Example request**

```

{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "anagrpId": "00103050h",
  "namespace": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "/vol/vol1/namespace1",
    "node": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "node1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "nsid": "00000001h",
  "subsystem": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "subsystem1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "svm": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  }
}

```

## Response

Status: 201, Created

Name	Type	Description
_links	<a href="#">_links</a>	
num_records	integer	The number of records in the response.
records	array[ <a href="#">nvme_subsystem_map</a> ]	

## Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": 1,
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "anagrp_id": "00103050h",
      "namespace": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "/vol/vol1/namespace1",
        "node": {
          "_links": {
            "self": {
              "href": "/api/resourcelink"
            }
          },
          "name": "node1",
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
        },
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      },
      "nsid": "00000001h",
      "subsystem": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "subsystem1",
```

```

    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "svm": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  }
}
]
}

```

## Headers

Name	Description	Type
Location	Useful for tracking the resource location	string

## Error

Status: Default

## ONTAP Error Response Codes

Error Code	Description
5374127	The specified namespace name is invalid.
5376461	The specified namespace name is invalid.
5376462	The specified namespace name is too long.
72089731	The specified NVMe namespace does not exist.
72089790	The supplied NVMe namespace is already mapped to the supplied NVMe subsystem.
72089793	An NVMe namespace in a snapshot cannot be mapped.
72089799	The NVMe namespace is the destination of an ongoing restore operation and is inaccessible for I/O and management.
72089902	A node does not have an NVMe interface configured.

Error Code	Description
72089903	Multiple nodes do not have an NVMe interface configured.
72089904	The aggregate must be given back to its home node prior to mapping the NVMe namespace it contains.
72090001	The NVMe subsystem specified by <code>subsystem.uuid</code> was not found.
72090005	The specified <code>namespace.uuid</code> and <code>namespace.name</code> refer to different NVMe namespaces.
72090006	The NVMe namespace specified by <code>namespace.uuid</code> was not found.
72090007	The NVMe namespace specified by <code>namespace.name</code> was not found.
72090020	The specified <code>subsystem.uuid</code> and <code>subsystem.name</code> refer to different NVMe subsystems.
72090021	The NVMe subsystem specified by <code>subsystem.name</code> was not found.

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

Name	Type	Description
error	<a href="#">returned_error</a>	

#### Example error

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



## See Definitions

href

Name	Type	Description
href	string	

\_links

Name	Type	Description
self	<a href="#">href</a>	

node

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	
uuid	string	

namespace

The NVMe namespace to which the NVMe subsystem is mapped. Required in POST by supplying either the UUID, name, or both.

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	The name of the NVMe namespace. Valid in POST. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.  NVMe namespace names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;namespace&gt;"</code> where the qtree name is optional.
node	<a href="#">node</a>	
uuid	string	The unique identifier of the NVMe namespace. Valid in POST.

subsystem

The NVMe subsystem to which the NVMe namespace is mapped. Required in POST by supplying either `subsystem.uuid`, `subsystem.name` or both.



Name	Type	Description
_links	<a href="#">_links</a>	
name	string	The name of the NVMe subsystem.
uuid	string	The unique identifier of the NVMe subsystem.

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
_links	<a href="#">_links</a>	
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\_map

An NVMe subsystem map is an association of an NVMe namespace with an NVMe subsystem. When an NVMe namespace is mapped to an NVMe subsystem, the NVMe subsystem's hosts are granted access to the NVMe namespace. The relationship between an NVMe subsystem and an NVMe namespace is one subsystem to many namespaces.

Name	Type	Description
_links	<a href="#">_links</a>	
anagrpId	string	<p>The Asymmetric Namespace Access Group ID (ANAGRPID) of the NVMe namespace.</p> <p>The format for an ANAGRPID is 8 hexadecimal digits (zero-filled) followed by a lower case "h".</p> <p>There is an added computational cost to retrieving this property's value. It is not populated for a GET request unless it is explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more.</p>

Name	Type	Description
namespace	<a href="#">namespace</a>	The NVMe namespace to which the NVMe subsystem is mapped. Required in POST by supplying either the UUID, name, or both.
nsid	string	<p>The NVMe namespace identifier. This is an identifier used by an NVMe controller to provide access to the NVMe namespace.</p> <p>The format for an NVMe namespace identifier is 8 hexadecimal digits (zero-filled) followed by a lower case "h".</p>
subsystem	<a href="#">subsystem</a>	The NVMe subsystem to which the NVMe namespace is mapped. Required in POST by supplying either <code>subsystem.uuid</code> , <code>subsystem.name</code> or both.
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.

#### links

Name	Type	Description
next	<a href="#">href</a>	
self	<a href="#">href</a>	

#### error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

#### returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	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 map

DELETE /protocols/nvme/subsystem-maps/{subsystem.uuid}/{namespace.uuid}

**Introduced In:** 9.6

Deletes an NVMe subsystem map.

### Related ONTAP commands

- `vserver nvme subsystem map remove`

### Learn more

- [DOC /protocols/nvme/subsystem-maps](#)

### Parameters

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

### Response

Status: 200, Ok

Name	Type	Description
_links	<a href="#">_links</a>	
num_records	integer	The number of records in the response.
records	array[ <a href="#">nvme_subsystem_map</a> ]	

## Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": 1,
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "anagrpId": "00103050h",
      "namespace": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "/vol/vol1/namespace1",
        "node": {
          "_links": {
            "self": {
              "href": "/api/resourcelink"
            }
          },
          "name": "node1",
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
        },
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      },
      "nsid": "00000001h",
      "subsystem": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "subsystem1",
```

```

    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "svm": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  }
}
]
}

```

## Error

Status: Default

## ONTAP Error Response Codes

Error Code	Description
72090019	The specified NVMe namespace is not mapped to the specified NVMe subsystem.

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

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	
self	<a href="#">href</a>	

\_links

Name	Type	Description
self	<a href="#">href</a>	

node

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	
uuid	string	

namespace

The NVMe namespace to which the NVMe subsystem is mapped. Required in POST by supplying either the UUID, name, or both.

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	<p>The name of the NVMe namespace. Valid in POST. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.</p> <p>NVMe namespace names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;namespace&gt;"</code> where the qtree name is optional.</p>
node	<a href="#">node</a>	

Name	Type	Description
uuid	string	The unique identifier of the NVMe namespace. Valid in POST.

#### subsystem

The NVMe subsystem to which the NVMe namespace is mapped. Required in POST by supplying either `subsystem.uuid`, `subsystem.name` or both.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
name	string	The name of the NVMe subsystem.
uuid	string	The unique identifier of the NVMe subsystem.

#### svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
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\_map

An NVMe subsystem map is an association of an NVMe namespace with an NVMe subsystem. When an NVMe namespace is mapped to an NVMe subsystem, the NVMe subsystem's hosts are granted access to the NVMe namespace. The relationship between an NVMe subsystem and an NVMe namespace is one subsystem to many namespaces.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	



Name	Type	Description
anagrpId	string	<p>The Asymmetric Namespace Access Group ID (ANAGRPID) of the NVMe namespace.</p> <p>The format for an ANAGRPID is 8 hexadecimal digits (zero-filled) followed by a lower case "h".</p> <p>There is an added computational cost to retrieving this property's value. It is not populated for a GET request unless it is explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more.</p>
namespace	<a href="#">namespace</a>	The NVMe namespace to which the NVMe subsystem is mapped. Required in POST by supplying either the UUID, name, or both.
nsid	string	<p>The NVMe namespace identifier. This is an identifier used by an NVMe controller to provide access to the NVMe namespace.</p> <p>The format for an NVMe namespace identifier is 8 hexadecimal digits (zero-filled) followed by a lower case "h".</p>
subsystem	<a href="#">subsystem</a>	The NVMe subsystem to which the NVMe namespace is mapped. Required in POST by supplying either <code>subsystem.uuid</code> , <code>subsystem.name</code> or both.
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.

#### error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

## Retrieve an NVMe subsystem map

GET /protocols/nvme/subsystem-maps/{subsystem.uuid}/{namespace.uuid}

**Introduced In:** 9.6

Retrieves an NVMe subsystem map.

### 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.

- `anagrpId`

### Related ONTAP commands

- `vserver nvme subsystem map show`

### Learn more

- [DOC /protocols/nvme/subsystem-maps](#)

### Parameters

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

Name	Type	In	Required	Description
fields	array[string]	query	False	Specify the fields to return.

## Response

Status: 200, Ok

Name	Type	Description
_links	<a href="#">_links</a>	
anagrpId	string	<p>The Asymmetric Namespace Access Group ID (ANAGRPID) of the NVMe namespace.</p> <p>The format for an ANAGRPID is 8 hexadecimal digits (zero-filled) followed by a lower case "h".</p> <p>There is an added computational cost to retrieving this property's value. It is not populated for a GET request unless it is explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more.</p>
namespace	<a href="#">namespace</a>	The NVMe namespace to which the NVMe subsystem is mapped. Required in POST by supplying either the UUID, name, or both.
nsid	string	<p>The NVMe namespace identifier. This is an identifier used by an NVMe controller to provide access to the NVMe namespace.</p> <p>The format for an NVMe namespace identifier is 8 hexadecimal digits (zero-filled) followed by a lower case "h".</p>
subsystem	<a href="#">subsystem</a>	The NVMe subsystem to which the NVMe namespace is mapped. Required in POST by supplying either <code>subsystem.uuid</code> , <code>subsystem.name</code> or both.

Name	Type	Description
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.

**Example response**

```

{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "anagrpId": "00103050h",
  "namespace": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "/vol/vol1/namespace1",
    "node": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "node1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "nsid": "00000001h",
  "subsystem": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "subsystem1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "svm": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  }
}

```

## Error

Status: Default

### ONTAP Error Response Codes

Error Code	Description
72090019	The specified NVMe namespace is not mapped to the specified NVMe subsystem.

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

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	

node

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	
uuid	string	

namespace

The NVMe namespace to which the NVMe subsystem is mapped. Required in POST by supplying either the UUID, name, or both.

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	The name of the NVMe namespace. Valid in POST. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.  NVMe namespace names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;namespace&gt;"</code> where the qtree name is optional.
node	<a href="#">node</a>	
uuid	string	The unique identifier of the NVMe namespace. Valid in POST.

subsystem

The NVMe subsystem to which the NVMe namespace is mapped. Required in POST by supplying either `subsystem.uuid`, `subsystem.name` or both.



Name	Type	Description
_links	<a href="#">_links</a>	
name	string	The name of the NVMe subsystem.
uuid	string	The unique identifier of the NVMe subsystem.

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
_links	<a href="#">_links</a>	
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
message	string	Message argument

returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

# 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": "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",
  "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 <ul style="list-style-type: none"><li>• Introduced in: 9.16</li></ul>
hosts.priority	string	query	False	Filter by hosts.priority <ul style="list-style-type: none"><li>• Introduced in: 9.14</li></ul>
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 <ul style="list-style-type: none"><li>• Introduced in: 9.12</li></ul>
hosts.dh_hmac_chap.mode	string	query	False	Filter by hosts.dh_hmac_chap.mode <ul style="list-style-type: none"><li>• Introduced in: 9.12</li></ul>
hosts.dh_hmac_chap.hash_function	string	query	False	Filter by hosts.dh_hmac_chap.hash_function <ul style="list-style-type: none"><li>• Introduced in: 9.12</li></ul>
vendor_uuids	string	query	False	Filter by vendor_uuids <ul style="list-style-type: none"><li>• Introduced in: 9.9</li></ul>

Name	Type	In	Required	Description
name	string	query	False	Filter by name <ul style="list-style-type: none"> <li>• maxLength: 64</li> <li>• minLength: 1</li> </ul>
serial_number	string	query	False	Filter by serial_number <ul style="list-style-type: none"> <li>• maxLength: 20</li> <li>• minLength: 20</li> </ul>
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 <ul style="list-style-type: none"> <li>• Introduced in: 9.7</li> </ul>
comment	string	query	False	Filter by comment <ul style="list-style-type: none"> <li>• maxLength: 255</li> <li>• minLength: 0</li> </ul>
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 <ul style="list-style-type: none"> <li>• maxLength: 223</li> <li>• minLength: 1</li> </ul>

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 <ul style="list-style-type: none"> <li>• Max value: 15</li> <li>• Min value: 1</li> </ul>
io_queue.default.depth	integer	query	False	Filter by io_queue.default.depth <ul style="list-style-type: none"> <li>• Max value: 128</li> <li>• Min value: 16</li> </ul>
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. <ul style="list-style-type: none"> <li>• Default value: 1</li> </ul>

Name	Type	In	Required	Description
return_timeout	integer	query	False	<p>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.</p> <ul style="list-style-type: none"> <li>• Max value: 120</li> <li>• Min value: 0</li> <li>• Default value: 1</li> </ul>
order_by	array[string]	query	False	Order results by specified fields and optional [asc

## Response

Status: 200, Ok

Name	Type	Description
_links	<a href="#">_links</a>	
num_records	integer	The number of records in the response.
records	array[ <a href="#">nvme_subsystem</a> ]	

## 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": {
            "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": [
    {
      "_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	<a href="#">returned_error</a>	

## 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	<a href="#">href</a>	
self	<a href="#">href</a>	

\_links

Name	Type	Description
self	<a href="#">href</a>	

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	<p>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.</p> <p>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.</p>

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.
host_secret_key	string	<p>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.</p> <p>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.</p>

Name	Type	Description
mode	string	<p>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:</p> <ul style="list-style-type: none"> <li>• none: The host has neither the host nor controller secret configured, and no authentication is performed.</li> <li>• unidirectional: The host has a host secret configured. The controller will authenticate the host.</li> <li>• bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.</li> </ul>

tls

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

Name	Type	Description
configured_psk	string	<p>A user supplied pre-shared key (PSK) value in PSK Interchange Format. Optional in POST.</p> <p>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 configured.</p> <p>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.</p>

Name	Type	Description
key_type	string	<p>The method by which the TLS pre-shared key (PSK) is configured for the host. Optional in POST.</p> <p>The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• <code>none</code> - TLS is not configured for the host connection. No value is allowed for property <code>configured_psk</code>.</li> <li>• <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.</li> </ul> <p>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>.</p>

#### hosts

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

## svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
_links	<a href="#">_links</a>	
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	<a href="#">_links</a>	



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[ <a href="#">hosts</a> ]	The NVMe hosts configured for access to the NVMe subsystem. Optional in POST.
io_queue	<a href="#">io_queue</a>	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[ <a href="#">subsystem_maps</a> ]	<p>The NVMe namespaces mapped to the NVMe subsystem.</p> <p>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 <a href="#">Requesting specific fields</a> to learn more.</p>
svm	<a href="#">svm</a>	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]	<p>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.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.9</li> <li>• readCreate: 1</li> </ul>

#### error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

#### returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	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. <ul style="list-style-type: none"> <li>• Default value:</li> </ul>

### Request Body

Name	Type	Description
_links	<a href="#">_links</a>	
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[ <a href="#">hosts</a> ]	The NVMe hosts configured for access to the NVMe subsystem. Optional in POST.
io_queue	<a href="#">io_queue</a>	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[ <a href="#">subsystem_maps</a> ]	<p>The NVMe namespaces mapped to the NVMe subsystem.</p> <p>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 <a href="#">Requesting specific fields</a> to learn more.</p>
svm	<a href="#">svm</a>	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]	<p>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.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.9</li> <li>• readCreate: 1</li> </ul>

## Example request

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "comment": "string",
  "hosts": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "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": [
    {
      "_links": {
```

```

    "self": {
      "href": "/api/resourcelink"
    },
    "anagrpid": "00103050h",
    "namespace": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "/vol/vol1/namespacel",
      "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"
  ]
}

```

## Response

Status: 201, Created

Name	Type	Description
_links	<a href="#">_links</a>	
num_records	integer	The number of records in the response.
records	array[ <a href="#">nvme_subsystem</a> ]	

## 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": {
            "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": [
    {
      "_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"
  ]
}
]
}

```

## 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.
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> .
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.

Error Code	Description
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.

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	

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	<p>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.</p> <p>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.</p>
group_size	string	<p>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.</p>

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	<p>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.</p> <p>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.</p>

Name	Type	Description
mode	string	<p>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:</p> <ul style="list-style-type: none"> <li>• none: The host has neither the host nor controller secret configured, and no authentication is performed.</li> <li>• unidirectional: The host has a host secret configured. The controller will authenticate the host.</li> <li>• bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.</li> </ul>

tls

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

Name	Type	Description
configured_psk	string	<p>A user supplied pre-shared key (PSK) value in PSK Interchange Format. Optional in POST.</p> <p>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>.</p> <p>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.</p>



Name	Type	Description
key_type	string	<p>The method by which the TLS pre-shared key (PSK) is configured for the host. Optional in POST.</p> <p>The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• <code>none</code> - TLS is not configured for the host connection. No value is allowed for property <code>configured_psk</code>.</li> <li>• <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.</li> </ul> <p>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>.</p>

#### hosts

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

## svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
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
<a href="#">_links</a>	<a href="#">_links</a>	

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[ <a href="#">hosts</a> ]	The NVMe hosts configured for access to the NVMe subsystem. Optional in POST.
io_queue	<a href="#">io_queue</a>	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[ <a href="#">subsystem_maps</a> ]	<p>The NVMe namespaces mapped to the NVMe subsystem.</p> <p>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 <a href="#">Requesting specific fields</a> to learn more.</p>
svm	<a href="#">svm</a>	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]	<p>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.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.9</li> <li>• readCreate: 1</li> </ul>

#### \_links

Name	Type	Description
next	<a href="#">href</a>	
self	<a href="#">href</a>	

#### error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

#### returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	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 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.

Name	Type	In	Required	Description
max_records	integer	query	False	Limit the number of records returned.
return_records	boolean	query	False	<p>The default is true for GET calls. When set to false, only the number of records is returned.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> </ul>
return_timeout	integer	query	False	<p>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.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> <li>• Max value: 120</li> <li>• Min value: 0</li> </ul>
order_by	array[string]	query	False	Order results by specified fields and optional [asc

## Response

Status: 200, Ok

Name	Type	Description
_links	<a href="#">_links</a>	
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": {
        "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": [
        {
          "_links": {
            "self": {
              "href": "/api/resourcelink"
            }
          },
          "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": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "name": "subsystem1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "tls": {
        "configured_psk": "NVMeTLSkey-
1:01:VRLbtnN9AQb2WXW3c9+wEf/DRLz0QuLdbYvEhwtdWwNf9LrZ:",
        "key_type": "configured"
    }
}
],
"subsystem": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "name": "subsystem1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"tls": {
    "configured_psk": "NVMeTLSkey-
1:01:VRLbtnN9AQb2WXW3c9+wEf/DRLz0QuLdbYvEhwtdWwNf9LrZ:",
    "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	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	
self	<a href="#">href</a>	

\_links

Name	Type	Description
self	<a href="#">href</a>	

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	<p>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.</p> <p>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.</p>

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.
host_secret_key	string	<p>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.</p> <p>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.</p>

Name	Type	Description
mode	string	<p>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:</p> <ul style="list-style-type: none"> <li>• none: The host has neither the host nor controller secret configured, and no authentication is performed.</li> <li>• unidirectional: The host has a host secret configured. The controller will authenticate the host.</li> <li>• bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.</li> </ul>

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	<a href="#">_links</a>	
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	<p>A user supplied pre-shared key (PSK) value in PSK Interchange Format. Optional in POST.</p> <p>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>.</p> <p>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.</p>

Name	Type	Description
key_type	string	<p>The method by which the TLS pre-shared key (PSK) is configured for the host. Optional in POST.</p> <p>The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• <code>none</code> - TLS is not configured for the host connection. No value is allowed for property <code>configured_psk</code>.</li> <li>• <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.</li> </ul> <p>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>.</p>

## records

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

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
<code>dh_hmac_chap</code>	<a href="#">dh_hmac_chap</a>	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.
<code>io_queue</code>	<a href="#">io_queue</a>	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	<a href="#">subsystem</a>	The NVMe subsystem to which the NVMe host has been provisioned.
tls	<a href="#">tls</a>	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	<a href="#">_links</a>	
dh_hmac_chap	<a href="#">dh_hmac_chap</a>	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.
io_queue	<a href="#">io_queue</a>	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[ <a href="#">records</a> ]	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	<a href="#">subsystem</a>	The NVMe subsystem to which the NVMe host has been provisioned.
tls	<a href="#">tls</a>	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[ <a href="#">error_arguments</a> ]	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
_links	<a href="#">_links</a>	
dh_hmac_chap	<a href="#">dh_hmac_chap</a>	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.
io_queue	<a href="#">io_queue</a>	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 <a href="#">io_queue</a> count and I/O queue depth are being used.
records	array[ <a href="#">records</a> ]	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	<a href="#">subsystem</a>	The NVMe subsystem to which the NVMe host has been provisioned.
tls	<a href="#">tls</a>	A container for the configuration for NVMe/TCP-TLS transport session for the host.

## Example request

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "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": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "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": {
        "_links": {
```

```

        "self": {
            "href": "/api/resourcelink"
        },
        "name": "subsystem1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "tls": {
        "configured_psk": "NVMeTLSkey-
1:01:VRLbtnN9AQb2WXW3c9+wEf/DRLz0QuLdbYvEhwtdWwNf9LrZ:",
        "key_type": "configured"
    }
},
"subsystem": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "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
_links	<a href="#">_links</a>	
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": {
        "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": [
        {
          "_links": {
            "self": {
              "href": "/api/resourcelink"
            }
          },
          "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": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "name": "subsystem1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "tls": {
        "configured_psk": "NVMeTLSkey-
1:01:VRLbtnN9AQb2WXW3c9+wEf/DRLz0QuLdbYvEhwtdWwNf9LrZ:",
        "key_type": "configured"
    }
}
],
"subsystem": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "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 '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.
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.

Error Code	Description
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.

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	

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	<p>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.</p> <p>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.</p>
group_size	string	<p>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.</p>

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	<p>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.</p> <p>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.</p>

Name	Type	Description
mode	string	<p>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:</p> <ul style="list-style-type: none"> <li>• none: The host has neither the host nor controller secret configured, and no authentication is performed.</li> <li>• unidirectional: The host has a host secret configured. The controller will authenticate the host.</li> <li>• bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.</li> </ul>

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
<a href="#">_links</a>	<a href="#">_links</a>	
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	<p>A user supplied pre-shared key (PSK) value in PSK Interchange Format. Optional in POST.</p> <p>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>.</p> <p>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.</p>

Name	Type	Description
key_type	string	<p>The method by which the TLS pre-shared key (PSK) is configured for the host. Optional in POST.</p> <p>The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• <code>none</code> - TLS is not configured for the host connection. No value is allowed for property <code>configured_psk</code>.</li> <li>• <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.</li> </ul> <p>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>.</p>

records

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

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
<code>dh_hmac_chap</code>	<a href="#">dh_hmac_chap</a>	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.
<code>io_queue</code>	<a href="#">io_queue</a>	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	<a href="#">subsystem</a>	The NVMe subsystem to which the NVMe host has been provisioned.
tls	<a href="#">tls</a>	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	<a href="#">_links</a>	
dh_hmac_chap	<a href="#">dh_hmac_chap</a>	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.
io_queue	<a href="#">io_queue</a>	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[ <a href="#">records</a> ]	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	<a href="#">subsystem</a>	The NVMe subsystem to which the NVMe host has been provisioned.
tls	<a href="#">tls</a>	A container for the configuration for NVMe/TCP-TLS transport session for the host.

\_links

Name	Type	Description
next	<a href="#">href</a>	
self	<a href="#">href</a>	

error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	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 ' <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> .
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	<a href="#">returned_error</a>	

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[ <a href="#">error_arguments</a> ]	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	<a href="#">_links</a>	
dh_hmac_chap	<a href="#">dh_hmac_chap</a>	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.
io_queue	<a href="#">io_queue</a>	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.
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.
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": {
    "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": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "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": {
        "_links": {
```

```

        "self": {
            "href": "/api/resourcelink"
        },
        "name": "subsystem1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "tls": {
        "configured_psk": "NVMeTLSkey-
1:01:VRLbtnN9AQb2WXW3c9+wEf/DRLz0QuLdbYvEhwtdWwNf9LrZ:",
        "key_type": "configured"
    }
},
"subsystem": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "name": "subsystem1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"tls": {
    "configured_psk": "NVMeTLSkey-
1:01:VRLbtnN9AQb2WXW3c9+wEf/DRLz0QuLdbYvEhwtdWwNf9LrZ:",
    "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.
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	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	

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	<p>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.</p> <p>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.</p>
group_size	string	<p>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.</p>

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	<p>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.</p> <p>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.</p>

Name	Type	Description
mode	string	<p>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:</p> <ul style="list-style-type: none"> <li>• none: The host has neither the host nor controller secret configured, and no authentication is performed.</li> <li>• unidirectional: The host has a host secret configured. The controller will authenticate the host.</li> <li>• bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.</li> </ul>

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
<a href="#">_links</a>	<a href="#">_links</a>	
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	<p>A user supplied pre-shared key (PSK) value in PSK Interchange Format. Optional in POST.</p> <p>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>.</p> <p>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.</p>

Name	Type	Description
key_type	string	<p>The method by which the TLS pre-shared key (PSK) is configured for the host. Optional in POST.</p> <p>The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• <code>none</code> - TLS is not configured for the host connection. No value is allowed for property <code>configured_psk</code>.</li> <li>• <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.</li> </ul> <p>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>.</p>

## records

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

Name	Type	Description
_links	<a href="#">_links</a>	
dh_hmac_chap	<a href="#">dh_hmac_chap</a>	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.
io_queue	<a href="#">io_queue</a>	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	<a href="#">subsystem</a>	The NVMe subsystem to which the NVMe host has been provisioned.
tls	<a href="#">tls</a>	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[ <a href="#">error_arguments</a> ]	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.
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	<a href="#">returned_error</a>	

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[ <a href="#">error_arguments</a> ]	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
<code>_links</code>	<a href="#">_links</a>	
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[ <a href="#">hosts</a> ]	The NVMe hosts configured for access to the NVMe subsystem. Optional in POST.
io_queue	<a href="#">io_queue</a>	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[ <a href="#">subsystem_maps</a> ]	<p>The NVMe namespaces mapped to the NVMe subsystem.</p> <p>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 <a href="#">Requesting specific fields</a> to learn more.</p>
svm	<a href="#">svm</a>	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]	<p>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.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.9</li> <li>• readCreate: 1</li> </ul>

## Example response

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "comment": "string",
  "hosts": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "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": [
    {
      "_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	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	

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	<p>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.</p> <p>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.</p>
group_size	string	<p>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.</p>

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	<p>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.</p> <p>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.</p>

Name	Type	Description
mode	string	<p>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:</p> <ul style="list-style-type: none"> <li>• none: The host has neither the host nor controller secret configured, and no authentication is performed.</li> <li>• unidirectional: The host has a host secret configured. The controller will authenticate the host.</li> <li>• bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.</li> </ul>

tls

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

Name	Type	Description
configured_psk	string	<p>A user supplied pre-shared key (PSK) value in PSK Interchange Format. Optional in POST.</p> <p>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>.</p> <p>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.</p>

Name	Type	Description
key_type	string	<p>The method by which the TLS pre-shared key (PSK) is configured for the host. Optional in POST.</p> <p>The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• <code>none</code> - TLS is not configured for the host connection. No value is allowed for property <code>configured_psk</code>.</li> <li>• <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.</li> </ul> <p>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>.</p>

#### hosts

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
<code>dh_hmac_chap</code>	<a href="#">dh_hmac_chap</a>	A container for the configuration of NVMe in-band authentication using the DH-HMAC-CHAP protocol for a host.
<code>nqn</code>	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	<a href="#">tls</a>	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	<a href="#">default</a>	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	<a href="#">_links</a>	
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
<a href="#">_links</a>	<a href="#">_links</a>	
anagrpId	string	<p>The Asymmetric Namespace Access Group ID (ANAGRPID) of the NVMe namespace.</p> <p>The format for an ANAGRPI is 8 hexadecimal digits (zero-filled) followed by a lower case "h".</p>
namespace	<a href="#">namespace</a>	An NVMe namespace mapped to the NVMe subsystem.
nsid	string	<p>The NVMe namespace identifier. This is an identifier used by an NVMe controller to provide access to the NVMe namespace.</p> <p>The format for an NVMe namespace identifier is 8 hexadecimal digits (zero-filled) followed by a lower case "h".</p>

## svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
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[ <a href="#">error_arguments</a> ]	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
_links	<a href="#">_links</a>	



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[ <a href="#">hosts</a> ]	The NVMe hosts configured for access to the NVMe subsystem. Optional in POST.
io_queue	<a href="#">io_queue</a>	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[ <a href="#">subsystem_maps</a> ]	<p>The NVMe namespaces mapped to the NVMe subsystem.</p> <p>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 <a href="#">Requesting specific fields</a> to learn more.</p>
svm	<a href="#">svm</a>	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]	<p>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.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.9</li> <li>• readCreate: 1</li> </ul>

## Example request

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "comment": "string",
  "hosts": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "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": [
    {
      "_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"
  ]
}

```

## 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.

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	

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	<p>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.</p> <p>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.</p>
group_size	string	<p>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.</p>

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	<p>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.</p> <p>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.</p>

Name	Type	Description
mode	string	<p>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:</p> <ul style="list-style-type: none"> <li>• none: The host has neither the host nor controller secret configured, and no authentication is performed.</li> <li>• unidirectional: The host has a host secret configured. The controller will authenticate the host.</li> <li>• bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.</li> </ul>

tls

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



Name	Type	Description
configured_psk	string	<p>A user supplied pre-shared key (PSK) value in PSK Interchange Format. Optional in POST.</p> <p>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>.</p> <p>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.</p>

Name	Type	Description
key_type	string	<p>The method by which the TLS pre-shared key (PSK) is configured for the host. Optional in POST.</p> <p>The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• <code>none</code> - TLS is not configured for the host connection. No value is allowed for property <code>configured_psk</code>.</li> <li>• <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.</li> </ul> <p>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>.</p>

#### hosts

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

## svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
_links	<a href="#">_links</a>	
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	<a href="#">_links</a>	

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[ <a href="#">hosts</a> ]	The NVMe hosts configured for access to the NVMe subsystem. Optional in POST.
io_queue	<a href="#">io_queue</a>	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[ <a href="#">subsystem_maps</a> ]	<p>The NVMe namespaces mapped to the NVMe subsystem.</p> <p>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 <a href="#">Requesting specific fields</a> to learn more.</p>
svm	<a href="#">svm</a>	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]	<p>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.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.9</li> <li>• readCreate: 1</li> </ul>

#### error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

#### returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code

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

## Manage NVMe namespaces

### Storage namespaces endpoint overview

#### Overview

An NVMe namespace is a collection of addressable logical blocks presented to hosts connected to the storage virtual machine using the NVMe over Fabrics protocol.

The NVMe namespace REST API allows you to create, update, delete and discover NVMe namespaces.

An NVMe namespace must be mapped to an NVMe subsystem to grant access to the subsystem's hosts. Hosts can then access the NVMe namespace and perform I/O using the NVMe over Fabrics protocol.

See the NVMe namespace object model to learn more about each of the properties supported by the NVMe namespace REST API. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.

NVMe namespace names are paths of the form `"/vol/<volume>[/<qtree>]/<namespace>"` where the qtree name is optional.

An NVMe namespace is created to a specified size using thin or thick provisioning as determined by the volume on which it is created. An NVMe namespace can then be resized or cloned. An NVMe namespace cannot be renamed, or moved to a different volume. NVMe namespaces do not support the assignment of a QoS policy for performance management, but a QoS policy can be assigned to the volume containing the namespace.

#### Performance monitoring

Performance of an NVMe namespace can be monitored by observing the `metric.*` and `statistics.*` properties. These properties show the space utilization and performance of an NVMe namespace in terms of IOPS, latency, and throughput. The `metric.*` properties denote an average, whereas `statistics.*` properties denote a real-time monotonically increasing value aggregated across all nodes.

#### Examples

##### Creating an NVMe namespace

This example creates a 300 gigabyte NVMe namespace, with 4096-byte blocks, in SVM `svm1`, volume `vol1`, configured for use by `linux` hosts. The `return_records` query parameter is used to retrieve properties of the newly created NVMe namespace in the POST response.

```
# The API:
```

```
POST /api/storage/namespaces
```

```
# The call:
```

```
curl -X POST 'https://<mgmt-  
ip>/api/storage/namespaces?return_records=true' -H 'Accept:  
application/hal+json' -d '{ "svm": { "name": "svm1" }, "os_type": "linux",  
"space": { "block_size": "4096", "size": "300G" }, "name" :  
"/vol/vol1/namespace1" }'
```

```
# The response:
```

```
{  
  "num_records": 1,  
  "records": [  
    {  
      "uuid": "dccc3e6-cf4e-498f-bec6-f7897f945669",  
      "svm": {  
        "uuid": "6bf967fd-2a1c-11e9-b682-005056bbc17d",  
        "name": "svm1",  
        "_links": {  
          "self": {  
            "href": "/api/svm/svms/6bf967fd-2a1c-11e9-b682-005056bbc17d"  
          }  
        }  
      },  
      "name": "/vol/vol1/namespace1",  
      "location": {  
        "namespace": "namespace1",  
        "volume": {  
          "uuid": "71cd0dba-2a1c-11e9-b682-005056bbc17d",  
          "name": "vol1",  
          "_links": {  
            "self": {  
              "href": "/api/storage/volumes/71cd0dba-2a1c-11e9-b682-  
005056bbc17d"  
            }  
          }  
        }  
      },  
      "enabled": true,  
      "os_type": "linux",  
      "space": {  
        "block_size": 4096,  
        "size": 322122547200,  
        "used": 0,  
        "guarantee": {  
          "requested": false,  

```



```
        "reserved": false
    },
    "status": {
        "container_state": "online",
        "read_only": false,
        "state": "online"
    },
    "_links": {
        "self": {
            "href": "/api/storage/namespaces/dccdc3e6-cf4e-498f-bec6-f7897f945669"
        }
    }
}
]
```

---

### Updating an NVMe namespace comment

This example sets the `comment` property of an NVMe namespace.

```
# The API:
PATCH /api/storage/namespaces/{uuid}

# The call:
```

### Updating the size of an NVMe namespace

This example increases the size of an NVMe namespace.

```
# The API:
PATCH /api/storage/namespaces/{uuid}

# The call:
curl -X PATCH 'https://<mgmt-ip>/api/storage/namespaces/dccdc3e6-cf4e-498f-bec6-f7897f945669' -H 'Accept: application/hal+json' -d '{ "space": { "size": "1073741824" } }'
```

## Retrieving NVMe namespaces

This example retrieves summary information for all online NVMe namespaces in SVM *svm1*. The `svm.name` and `status.state` query parameters are to find the desired NVMe namespaces.

```
# The API:
GET /api/storage/namespaces

# The call:
curl -X GET 'https://<mgmt-
ip>/api/storage/namespaces?svm.name=svm1&status.state=online' -H 'Accept:
application/hal+json'

# The response:
{
  "records": [
    {
      "uuid": "5c254d22-96a6-42ac-aad8-0cd9ebd126b6",
      "svm": {
        "name": "svm1"
      },
      "name": "/vol/vol1/namespace2",
      "status": {
        "state": "online"
      },
      "_links": {
        "self": {
          "href": "/api/storage/namespaces/5c254d22-96a6-42ac-aad8-
0cd9ebd126b6"
        }
      },
    },
    {
      "uuid": "dccdc3e6-cf4e-498f-bec6-f7897f945669",
      "svm": {
        "name": "svm1"
      },
      "name": "/vol/vol1/namespace1",
      "status": {
        "state": "online"
      },
      "_links": {
        "self": {
          "href": "/api/storage/namespaces/dccdc3e6-cf4e-498f-bec6-
f7897f945669"
        }
      },
    }
  ]
}
```

```

    }
  },
  {
    "uuid": "be732687-20cf-47d2-a0e2-2a989d15661d",
    "svm": {
      "name": "svm1"
    },
    "name": "/vol/vol2/namespace3",
    "status": {
      "state": "online"
    },
    "_links": {
      "self": {
        "href": "/api/storage/namespaces/be732687-20cf-47d2-a0e2-2a989d15661d"
      }
    }
  }
],
"num_records": 3,
"_links": {
  "self": {
    "href": "/api/storage/namespaces?svm.name=svm1&status.state=online"
  }
}
}

```

### Retrieving details for a specific NVMe namespace

In this example, the `fields` query parameter is used to request all fields, including advanced fields, that would not otherwise be returned by default for the NVMe namespace.

```

# The API:
GET /api/storage/namespaces/{uuid}

# The call:
curl -X GET 'https://<mgmt-ip>/api/storage/namespaces/dccdc3e6-cf4e-498f-bec6-f7897f945669?fields=**' -H 'Accept: application/hal+json'

# The response:
{
  "uuid": "dccdc3e6-cf4e-498f-bec6-f7897f945669",
  "svm": {
    "uuid": "6bf967fd-2a1c-11e9-b682-005056bbc17d",

```

```

    "name": "svm1",
    "_links": {
      "self": {
        "href": "/api/svm/svms/6bf967fd-2a1c-11e9-b682-005056bbc17d"
      }
    }
  },
  "name": "/vol/vol1/namespace1",
  "location": {
    "namespace": "namespace1",
    "volume": {
      "uuid": "71cd0dba-2a1c-11e9-b682-005056bbc17d",
      "name": "vol1",
      "_links": {
        "self": {
          "href": "/api/storage/volumes/71cd0dba-2a1c-11e9-b682-005056bbc17d"
        }
      }
    }
  },
  "auto_delete": false,
  "enabled": true,
  "comment": "Data for the research department.",
  "os_type": "linux",
  "space": {
    "block_size": 4096,
    "size": 322122547200,
    "used": 0,
    "guarantee": {
      "requested": false,
      "reserved": false
    }
  },
  "status": {
    "container_state": "online",
    "mapped": true,
    "read_only": false,
    "state": "online"
  },
  "subsystem_map": {
    "nsid": "00000001h",
    "anagrpid": "00000001h",
    "subsystem": {
      "uuid": "01f17d05-2be9-11e9-bed2-005056bbc17d",
      "name": "subsystem1",

```

```

    "_links": {
      "self": {
        "href": "/api/protocols/nvme/subsystems/01f17d05-2be9-11e9-bed2-005056bbc17d"
      }
    },
    "_links": {
      "self": {
        "href": "/api/protocols/nvme/subsystem-maps/dccdc3e6-cf4e-498f-bec6-f7897f945669/01f17d05-2be9-11e9-bed2-005056bbc17d"
      }
    }
  },
  "metric": {
    "timestamp": "2019-04-09T05:50:15Z",
    "duration": "PT15S",
    "status": "ok",
    "latency": {
      "other": 0,
      "total": 0,
      "read": 0,
      "write": 0
    },
    "iops": {
      "read": 0,
      "write": 0,
      "other": 0,
      "total": 0
    },
    "throughput": {
      "read": 0,
      "write": 0,
      "total": 0
    }
  },
  "statistics": {
    "timestamp": "2019-04-09T05:50:42Z",
    "status": "ok",
    "latency_raw": {
      "other": 38298,
      "total": 38298,
      "read": 0,
      "write": 0
    },
    "iops_raw": {

```

```

    "read": 0,
    "write": 0,
    "other": 3,
    "total": 3
  },
  "throughput_raw": {
    "read": 0,
    "write": 0,
    "total": 0
  }
},
"_links": {
  "self": {
    "href": "/api/storage/namespaces/dccdc3e6-cf4e-498f-bec6-f7897f945669?fields=**"
  }
}
}

```

## Cloning NVMe namespaces

A clone of an NVMe namespace is an independent "copy" of the namespace that shares unchanged data blocks with the original. As blocks of the source and clone are modified, unique blocks are written for each. NVMe namespace clones can be created quickly and consume very little space initially. They can be created for the purpose of back-up, or to replicate data for multiple consumers.

An NVMe namespace clone can also be set to auto-delete by setting the `auto_delete` property. If the namespace's volume is configured for automatic deletion, NVMe namespaces that have auto-delete enabled are deleted when a volume is nearly full to reclaim a target amount of free space in the volume.

### Creating a new NVMe namespace clone

You create an NVMe namespace clone as you create any NVMe namespace — a POST to [/storage/namespaces](#). Set `clone.source.uuid` or `clone.source.name` to identify the source NVMe namespace from which the clone is created. The NVMe namespace clone and its source must reside in the same volume.

The source NVMe namespace can reside in a snapshot, in which case, the `clone.source.name` field must be used to identify it. Add `/.snapshot/<snapshot_name>` to the path after the volume name to identify the snapshot. For example `/vol/vol1/.snapshot/snap1/namespace1`.

```
# The API:
POST /api/storage/namespaces

# The call:
curl -X POST 'https://<mgmt-ip>/api/storage/namespaces' -H 'Accept:
application/hal+json' -d '{ "svm": { "name": "svm1" }, "name":
"/vol/vol1/namespace2clone1", "clone": { "source": { "name":
"/vol/vol1/namespace2" } } }'
```

### Over-writing an existing NVMe namespace's data as a clone of another

You can over-write an existing NVMe namespace as a clone of another. You do this as a PATCH on the NVMe namespace to overwrite — a PATCH to [/storage/namespaces/{uuid}](#). Set the `clone.source.uuid` or `clone.source.name` property to identify the source NVMe namespace from which the clone data is taken. The NVMe namespace clone and its source must reside in the same volume.

When used in a PATCH, the patched NVMe namespace's data is over-written as a clone of the source and the following properties are preserved from the patched namespace unless otherwise specified as part of the PATCH: `auto_delete`, `subsystem_map`, `status.state`, and `uuid`.

```
# The API:
PATCH /api/storage/namespaces/{uuid}

# The call:
curl -X PATCH 'https://<mgmt-ip>/api/storage/namespaces/dccdc3e6-cf4e-
498f-bec6-f7897f945669' -H 'Accept: application/hal+json' -d '{ "clone": {
"source": { "name": "/vol/vol1/namespace2" } } }'
```

### Converting a LUN into an NVMe namespace

An existing LUN can be converted in-place to an NVMe namespace with no modification to the data blocks. In other words, there is no additional copy created for the data blocks. There are certain requirements when converting a LUN to an NVMe namespace. For instance, the LUN should not be mapped to an initiator group, or exist as a protocol endpoint LUN, or in a foreign LUN import relationship. If the LUN exists as a VM volume, it should not be bound to a protocol endpoint LUN. Furthermore, only LUN with a supported operating system type for NVMe namespace can be converted.

The conversion process updates the metadata to the LUN, making it an NVMe namespace. The conversion is both time and space efficient. After conversion, the new namespace behaves as a regular namespace and may be mapped to an NVMe subsystem.

#### Convert a LUN into an NVMe namespace

You convert a LUN into an NVMe namespace by calling a POST to [/storage/namespaces](#). Set `convert.lun.uuid` or `convert.lun.name` to identify the source LUN which is to be converted in-place into

an NVMe namespace.

```
# The API:
POST /api/storage/namespaces

# The call:
curl -X POST 'https://<mgmt-ip>/api/storage/namespaces' -H 'Accept:
application/hal+json' -d '{ "svm": { "name": "svm1" }, "convert": { "lun":
{ "name": "/vol/vol1/lun1" } } }'
```

---

## Deleting an NVMe namespace

```
# The API:
DELETE /api/storage/namespaces/{uuid}

# The call:
curl -X DELETE 'https://<mgmt-ip>/api/storage/namespaces/5c254d22-96a6-
42ac-aad8-0cd9ebd126b6' -H 'Accept: application/hal+json'
```

---

## Retrieve NVMe namespaces

GET /storage/namespaces

**Introduced In:** 9.6

Retrieves NVMe namespaces.

### 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.

- `auto_delete`
- `space.physical_used`
- `space.physical_used_by_snapshots`
- `space.efficiency_ratio`
- `subsystem_map.*`
- `status.mapped`
- `statistics.*`



- `metric.*`

## Related ONTAP commands

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

## Learn more

- [DOC /storage/namespaces](#) to learn more and examples.

## Parameters

Name	Type	In	Required	Description
comment	string	query	False	Filter by comment <ul style="list-style-type: none"> <li>• maxLength: 254</li> <li>• minLength: 0</li> </ul>
subsystem_map.ana grpid	string	query	False	Filter by subsystem_map.ana grpid
subsystem_map.sub system.os_type	string	query	False	Filter by subsystem_map.sub system.os_type <ul style="list-style-type: none"> <li>• Introduced in: 9.16</li> </ul>
subsystem_map.sub system.uuid	string	query	False	Filter by subsystem_map.sub system.uuid
subsystem_map.sub system.comment	string	query	False	Filter by subsystem_map.sub system.comment <ul style="list-style-type: none"> <li>• maxLength: 255</li> <li>• minLength: 0</li> <li>• Introduced in: 9.16</li> </ul>

Name	Type	In	Required	Description
subsystem_map.subsystem.hosts.dh_hmac_chap.group_size	string	query	False	Filter by subsystem_map.subsystem.hosts.dh_hmac_chap.group_size <ul style="list-style-type: none"> <li>Introduced in: 9.16</li> </ul>
subsystem_map.subsystem.hosts.dh_hmac_chap.mode	string	query	False	Filter by subsystem_map.subsystem.hosts.dh_hmac_chap.mode <ul style="list-style-type: none"> <li>Introduced in: 9.16</li> </ul>
subsystem_map.subsystem.hosts.dh_hmac_chap.hash_function	string	query	False	Filter by subsystem_map.subsystem.hosts.dh_hmac_chap.hash_function <ul style="list-style-type: none"> <li>Introduced in: 9.16</li> </ul>
subsystem_map.subsystem.hosts.nqn	string	query	False	Filter by subsystem_map.subsystem.hosts.nqn <ul style="list-style-type: none"> <li>Introduced in: 9.16</li> </ul>
subsystem_map.subsystem.hosts.priority	string	query	False	Filter by subsystem_map.subsystem.hosts.priority <ul style="list-style-type: none"> <li>Introduced in: 9.16</li> </ul>
subsystem_map.subsystem.hosts.tls.key_type	string	query	False	Filter by subsystem_map.subsystem.hosts.tls.key_type <ul style="list-style-type: none"> <li>Introduced in: 9.16</li> </ul>

Name	Type	In	Required	Description
subsystem_map.subsystem.name	string	query	False	Filter by subsystem_map.subsystem.name <ul style="list-style-type: none"> <li>• maxLength: 64</li> <li>• minLength: 1</li> </ul>
subsystem_map.nsid	string	query	False	Filter by subsystem_map.nsid
os_type	string	query	False	Filter by os_type
auto_delete	boolean	query	False	Filter by auto_delete
status.mapped	boolean	query	False	Filter by status.mapped
status.read_only	boolean	query	False	Filter by status.read_only
status.state	string	query	False	Filter by status.state
status.container_state	string	query	False	Filter by status.container_state
uuid	string	query	False	Filter by uuid
space.physical_used	integer	query	False	Filter by space.physical_used <ul style="list-style-type: none"> <li>• Introduced in: 9.16</li> </ul>
space.guarantee.requested	boolean	query	False	Filter by space.guarantee.requested
space.guarantee.reserved	boolean	query	False	Filter by space.guarantee.reserved
space.block_size	integer	query	False	Filter by space.block_size

Name	Type	In	Required	Description
space.encyracy_rati o	number	query	False	Filter by space.encyracy_rati o  • Introduced in: 9.16
space.used	integer	query	False	Filter by space.used
space.size	integer	query	False	Filter by space.size  • Max value: 1407374883553 28  • Min value: 4096
space.physical_used _by_snapshots	integer	query	False	Filter by space.physical_use d_by_snapshots  • Introduced in: 9.16
create_time	string	query	False	Filter by create_time  • Introduced in: 9.7
statistics.iops_raw.w rite	integer	query	False	Filter by statistics.iops_raw.w rite  • Introduced in: 9.8
statistics.iops_raw.to tal	integer	query	False	Filter by statistics.iops_raw.to tal  • Introduced in: 9.8
statistics.iops_raw.re ad	integer	query	False	Filter by statistics.iops_raw.r ead  • Introduced in: 9.8

Name	Type	In	Required	Description
statistics.iops_raw.other	integer	query	False	Filter by statistics.iops_raw.other  • Introduced in: 9.8
statistics.throughput_raw.total	integer	query	False	Filter by statistics.throughput_raw.total  • Introduced in: 9.8
statistics.throughput_raw.write	integer	query	False	Filter by statistics.throughput_raw.write  • Introduced in: 9.8
statistics.throughput_raw.read	integer	query	False	Filter by statistics.throughput_raw.read  • Introduced in: 9.8
statistics.timestamp	string	query	False	Filter by statistics.timestamp  • Introduced in: 9.8
statistics.latency_raw.write	integer	query	False	Filter by statistics.latency_raw.write  • Introduced in: 9.8
statistics.latency_raw.total	integer	query	False	Filter by statistics.latency_raw.total  • Introduced in: 9.8

Name	Type	In	Required	Description
statistics.latency_raw.read	integer	query	False	Filter by statistics.latency_raw.read  • Introduced in: 9.8
statistics.latency_raw.other	integer	query	False	Filter by statistics.latency_raw.other  • Introduced in: 9.8
statistics.status	string	query	False	Filter by statistics.status  • Introduced in: 9.8
consistency_group.uuid	string	query	False	Filter by consistency_group.uuid  • Introduced in: 9.16
consistency_group.name	string	query	False	Filter by consistency_group.name  • Introduced in: 9.16
name	string	query	False	Filter by name
metric.status	string	query	False	Filter by metric.status  • Introduced in: 9.8
metric.timestamp	string	query	False	Filter by metric.timestamp  • Introduced in: 9.8

Name	Type	In	Required	Description
metric.iops.write	integer	query	False	Filter by metric.iops.write  • Introduced in: 9.8
metric.iops.total	integer	query	False	Filter by metric.iops.total  • Introduced in: 9.8
metric.iops.read	integer	query	False	Filter by metric.iops.read  • Introduced in: 9.8
metric.iops.other	integer	query	False	Filter by metric.iops.other  • Introduced in: 9.8
metric.latency.write	integer	query	False	Filter by metric.latency.write  • Introduced in: 9.8
metric.latency.total	integer	query	False	Filter by metric.latency.total  • Introduced in: 9.8
metric.latency.read	integer	query	False	Filter by metric.latency.read  • Introduced in: 9.8
metric.latency.other	integer	query	False	Filter by metric.latency.other  • Introduced in: 9.8

Name	Type	In	Required	Description
metric.duration	string	query	False	Filter by metric.duration <ul style="list-style-type: none"> <li>Introduced in: 9.8</li> </ul>
metric.throughput.total	integer	query	False	Filter by metric.throughput.total <ul style="list-style-type: none"> <li>Introduced in: 9.8</li> </ul>
metric.throughput.write	integer	query	False	Filter by metric.throughput.write <ul style="list-style-type: none"> <li>Introduced in: 9.8</li> </ul>
metric.throughput.read	integer	query	False	Filter by metric.throughput.read <ul style="list-style-type: none"> <li>Introduced in: 9.8</li> </ul>
svm.name	string	query	False	Filter by svm.name
svm.uuid	string	query	False	Filter by svm.uuid
location.volume.uuid	string	query	False	Filter by location.volume.uuid
location.volume.name	string	query	False	Filter by location.volume.name
location.namespace	string	query	False	Filter by location.namespace
location.node.uuid	string	query	False	Filter by location.node.uuid <ul style="list-style-type: none"> <li>Introduced in: 9.10</li> </ul>



Name	Type	In	Required	Description
location.node.name	string	query	False	Filter by location.node.name <ul style="list-style-type: none"> <li>• Introduced in: 9.10</li> </ul>
location.qtree.name	string	query	False	Filter by location.qtree.name
location.qtree.id	integer	query	False	Filter by location.qtree.id <ul style="list-style-type: none"> <li>• Max value: 4994</li> <li>• Min value: 0</li> </ul>
enabled	boolean	query	False	Filter by enabled
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. <ul style="list-style-type: none"> <li>• Default value: 1</li> </ul>

Name	Type	In	Required	Description
return_timeout	integer	query	False	<p>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.</p> <ul style="list-style-type: none"> <li>• Max value: 120</li> <li>• Min value: 0</li> <li>• Default value: 1</li> </ul>
order_by	array[string]	query	False	Order results by specified fields and optional [asc

## Response

Status: 200, Ok

Name	Type	Description
_links	<a href="#">_links</a>	
num_records	integer	The number of records in the response.
records	array[nvme_namespace]	

## Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": 1,
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "clone": {
        "source": {
          "name": "/vol/volume1/namespace1",
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
        }
      },
      "comment": "string",
      "consistency_group": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "cg1",
        "uuid": "4abc2317-4332-9d37-93a0-20bd29c22df0"
      },
      "convert": {
        "lun": {
          "name": "/vol/volume1/lun1",
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
        }
      },
      "create_time": "2018-06-04 15:00:00 -0400",
      "location": {
        "namespace": "namespace1",
        "node": {
          "_links": {
```

```

        "self": {
            "href": "/api/resourcelink"
        },
        "name": "node1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "qtree": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "id": 1,
        "name": "qt1"
    },
    "volume": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "name": "volume1",
        "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
    },
    "metric": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "duration": "PT15S",
        "iops": {
            "read": 200,
            "total": 1000,
            "write": 100
        },
        "latency": {
            "read": 200,
            "total": 1000,
            "write": 100
        },
        "status": "ok",
        "throughput": {
            "read": 200,

```

```

        "total": 1000,
        "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
},
"name": "/vol/volume1/qtree1/namespace1",
"os_type": "string",
"provisioning_options": {
    "qos_policy": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "name": "performance",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "snapshot_policy": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "name": "default",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "storage_service": {
        "name": "string"
    },
    "tiering": {
        "control": "string",
        "object_stores": [
            {
                "name": "string"
            }
        ],
        "policy": "string"
    }
},
"space": {
    "block_size": 512,
    "efficiency_ratio": 2.5,
    "physical_used": 1073741824,
    "physical_used_by_snapshots": 1073741824,
    "size": 1073741824,
    "used": 0
}

```

```

},
"statistics": {
  "iops_raw": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "latency_raw": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "status": "ok",
  "throughput_raw": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "timestamp": "2017-01-25 06:20:13 -0500"
},
"status": {
  "container_state": "string",
  "state": "online"
},
"subsystem_map": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "anagrpid": "00103050h",
  "nsid": "00000001h",
  "subsystem": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    }
  },
  "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"
    }
},
    "name": "subsystem1",
    "os_type": "string",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
},
"svm": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
},
"uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
]
}

```

## Error

Status: Default, Error

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	
self	<a href="#">href</a>	

\_links

Name	Type	Description
self	<a href="#">href</a>	

source

The source NVMe namespace for a namespace clone operation. This can be specified using property `clone.source.uuid` or `clone.source.name`. If both properties are supplied, they must refer to the same namespace.

Valid in POST to create a new NVMe namespace as a clone of the source.

Valid in PATCH to overwrite an existing NVMe namespace's data as a clone of another.

Name	Type	Description
name	string	<p>The name of the clone source NVMe namespace. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.</p> <p>NVMe namespace names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;namespace&gt;"</code> where the qtree name is optional.</p> <p>Valid in POST and PATCH.</p>
uuid	string	<p>The unique identifier of the clone source NVMe namespace. Valid in POST and PATCH.</p>

clone

This sub-object is used in POST to create a new NVMe namespace as a clone of an existing namespace, or PATCH to overwrite an existing namespace as a clone of another. Setting a property in this sub-object indicates that a namespace clone is desired.

When used in a PATCH, the patched NVMe namespace's data is over-written as a clone of the source and the following properties are preserved from the patched namespace unless otherwise specified as part of the PATCH: `auto_delete` (unless specified in the request), `subsystem_map`, `status.state`, and `uuid`.

Name	Type	Description
source	<a href="#">source</a>	<p>The source NVMe namespace for a namespace clone operation. This can be specified using property <code>clone.source.uuid</code> or <code>clone.source.name</code>. If both properties are supplied, they must refer to the same namespace.</p> <p>Valid in POST to create a new NVMe namespace as a clone of the source.</p> <p>Valid in PATCH to overwrite an existing NVMe namespace's data as a clone of another.</p>

#### consistency\_group

The namespace's consistency group. This property is populated for namespaces that are members of a consistency group. If the namespace is a member of a child consistency group, the parent consistency group is reported. A namespace's consistency group is the consistency group of its containing volume.

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
name	string	The name of the consistency group.
uuid	string	The unique identifier of the consistency group.

#### lun

The source LUN for convert operation. This can be specified using property `convert.lun.uuid` or `convert.lun.name`. If both properties are supplied, they must refer to the same LUN.

Valid in POST. A convert request from LUN to NVMe namespace cannot be combined with setting any other namespace properties. All other properties of the converted NVMe namespace come from the source LUN.

Name	Type	Description
name	string	The name of the source LUN. Valid in POST. A LUN is located within a volume. Optionally, it can be located within a qtree in a volume.  LUN names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;lun&gt;"</code> where the qtree name is optional.
uuid	string	The unique identifier of the source LUN. Valid in POST.

## convert

This sub-object is used in POST to convert a valid in-place LUN to an NVMe namespace. Setting a property in this sub-object indicates that a conversion from the specified LUN to NVMe namespace is desired.

Name	Type	Description
lun	<a href="#">lun</a>	The source LUN for convert operation. This can be specified using property <code>convert.lun.uuid</code> or <code>convert.lun.name</code> . If both properties are supplied, they must refer to the same LUN.  Valid in POST. A convert request from LUN to NVMe namespace cannot be combined with setting any other namespace properties. All other properties of the converted NVMe namespace come from the source LUN.

## node

The cluster node that hosts the NVMe namespace.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
name	string	
uuid	string	

## qtree

The qtree in which the NVMe namespace is optionally located. Valid in POST.

If properties `name` and `location.qtree.name` and/or `location.qtree.uuid` are specified in the same request, they must refer to the same qtree.

NVMe namespaces do not support rename.

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
<code>id</code>	integer	The identifier for the qtree, unique within the qtree's volume.
<code>name</code>	string	The name of the qtree.

volume

The volume in which the NVMe namespace is located. Valid in POST.

If properties `name` and `location.volume.name` and/or `location.volume.uuid` are specified in the same request, they must refer to the same volume.

NVMe namespaces do not support movement between volumes.

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
<code>name</code>	string	The name of the volume. This field cannot be specified in a PATCH method.
<code>uuid</code>	string	Unique identifier for the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move. <ul style="list-style-type: none"><li>• example: 028baa66-41bd-11e9-81d5-00a0986138f7</li><li>• Introduced in: 9.6</li><li>• x-nullable: true</li></ul>

location

The location of the NVMe namespace within the ONTAP cluster. NVMe namespaces do not support rename, or movement between volumes. Valid in POST.

Name	Type	Description
namespace	string	<p>The base name component of the NVMe namespace. Valid in POST.</p> <p>If properties <code>name</code> and <code>location.namespace</code> are specified in the same request, they must refer to the base name.</p> <p>NVMe namespaces do not support rename.</p>
node	<a href="#">node</a>	The cluster node that hosts the NVMe namespace.
qtree	<a href="#">qtree</a>	<p>The qtree in which the NVMe namespace is optionally located. Valid in POST.</p> <p>If properties <code>name</code> and <code>location.qtree.name</code> and/or <code>location.qtree.uuid</code> are specified in the same request, they must refer to the same qtree.</p> <p>NVMe namespaces do not support rename.</p>
volume	<a href="#">volume</a>	<p>The volume in which the NVMe namespace is located. Valid in POST.</p> <p>If properties <code>name</code> and <code>location.volume.name</code> and/or <code>location.volume.uuid</code> are specified in the same request, they must refer to the same volume.</p> <p>NVMe namespaces do not support movement between volumes.</p>

iops

The rate of I/O operations observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### latency

The round trip latency in microseconds observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### throughput

The rate of throughput bytes per second observed at the storage object.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.

Name	Type	Description
write	integer	Performance metric for write I/O operations.

metric

Performance numbers, such as IOPS latency and throughput

Name	Type	Description
_links	<a href="#">_links</a>	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
iops	<a href="#">iops</a>	The rate of I/O operations observed at the storage object.
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.

Name	Type	Description
throughput	<a href="#">throughput</a>	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

self\_link

Name	Type	Description
self	<a href="#">href</a>	

qos\_policy

The QoS policy for the volume provisioned to host the namespace. This property is only supported when the request provisions a new volume. If no `qos_policy` is provided, a policy is be set based on the `storage_service.name`, which defaults to the most performant service available.

Name	Type	Description
_links	<a href="#">self_link</a>	
name	string	The QoS policy group name. This is mutually exclusive with UUID and other QoS attributes during POST and PATCH.
uuid	string	The QoS policy group UUID. This is mutually exclusive with name and other QoS attributes during POST and PATCH.

snapshot\_policy\_reference

This is a reference to the snapshot policy.

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	
uuid	string	

storage\_service

Determines the placement of the namespace based on the value specified. Valid in POST.



Name	Type	Description
name	string	Storage service name. If not specified, the default value is the most performant for the platform.

#### object\_stores

Name	Type	Description
name	string	The name of the object store to use. Used for placement.

#### tiering

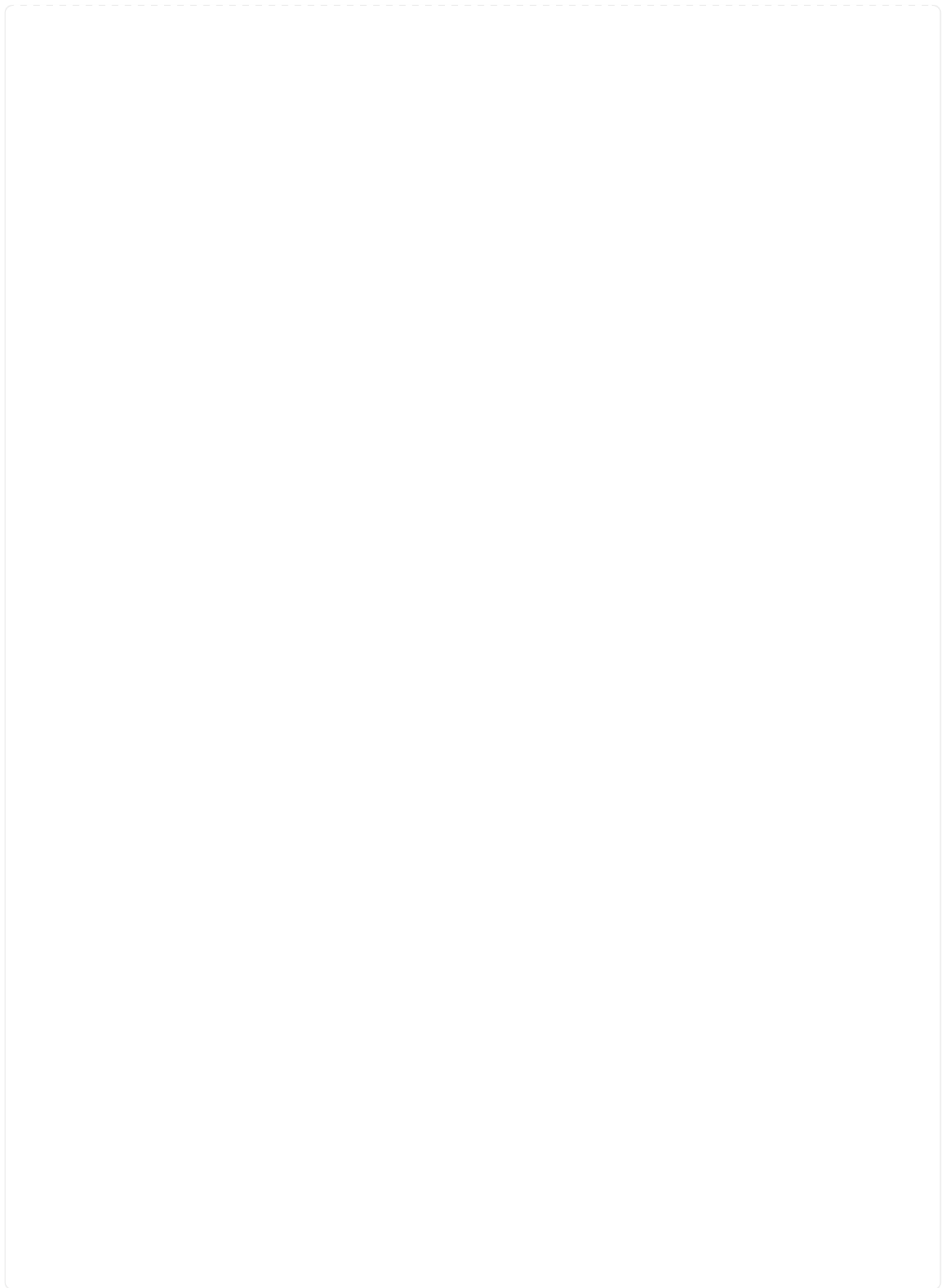
The tiering placement and policy definitions for the volume provisioned to host the namespace. This property is only supported when the request provisions a new volume.

Name	Type	Description
control	string	Storage tiering placement rules for the object.
object_stores	array[ <a href="#">object_stores</a> ]	Object stores to use. Used for placement.

Name	Type	Description
policy	string	<p>Policy that determines whether the user data blocks of a volume in a FabricPool will be tiered to the cloud store when they become cold.</p> <p>FabricPool combines flash (performance tier) with a cloud store into a single aggregate. Temperature of a volume block increases if it is accessed frequently and decreases when it is not. Valid in POST or PATCH.</p> <p>all &amp;dash; Allows tiering of both snapshots and active file system user data to the cloud store as soon as possible by ignoring the temperature on the volume blocks.</p> <p>auto &amp;dash; Allows tiering of both snapshot and active file system user data to the cloud store</p> <p>none &amp;dash; Volume blocks are not be tiered to the cloud store.</p> <p>snapshot_only &amp;dash; Allows tiering of only the volume snapshots not associated with the active file system.</p> <p>The default tiering policy is "snapshot-only" for a FlexVol volume and "none" for a FlexGroup volume. The default minimum cooling period for the "snapshot-only" tiering policy is 2 days and for the "auto" tiering policy it is 31 days.</p>

provisioning\_options

Options that are applied to the operation.



Name	Type	Description
auto	boolean	<p>If the volume specified in the request does not exist, automatically provision one of appropriate size. If the volume does exist, resize it to accommodate the new namespace.</p> <p>This property is only supported on Unified ONTAP.</p> <p>The following behavior is different from a traditional POST request:</p> <ul style="list-style-type: none"> <li>• The operation is asynchronous.</li> <li>• The <code>qos_policy</code> property is supported and is applied to the provisioned volume. A default QoS policy is applied to the volume if one is not provided.</li> <li>• The <code>provisioning_options.count</code> property is supported, provisioning <i>count</i> namespaces on the volume using the specified properties.</li> <li>• The <code>subsystem_map</code> property is supported. If the specified subsystem does not exist, it is created. The namespace is mapped to this subsystem. If a subsystem is provisioned in this way, it is deleted after it is no longer mapped to any namespaces.</li> <li>• The <code>clone</code> and <code>convert</code> properties are not supported.</li> <li>• When performing <code>records</code> based operations, specifying this property in the query applies to the entire operation. Specifying it for an individual record within the request applies to only that record.</li> <li>• Many other <code>provisioning_options</code> properties are supported to control the placement of the namespace and the properties of the volume</li> </ul>

Name	Type	Description
count	integer	The number of namespaces to provision with these properties. Only POST requests based on <code>space.size</code> are supported. When provided, the name is considered a prefix, and a suffix of the form <code>_&lt;N&gt;</code> is generated where N is the next available numeric index, starting with 1.
qos_policy	<a href="#">qos_policy</a>	The QoS policy for the volume provisioned to host the namespace. This property is only supported when the request provisions a new volume. If no <code>qos_policy</code> is provided, a policy is be set based on the <code>storage_service.name</code> , which defaults to the most performant service available.
snapshot_policy	<a href="#">snapshot_policy_reference</a>	This is a reference to the snapshot policy.
storage_service	<a href="#">storage_service</a>	Determines the placement of the namespace based on the value specified. Valid in POST.
tiering	<a href="#">tiering</a>	The tiering placement and policy definitions for the volume provisioned to host the namespace. This property is only supported when the request provisions a new volume.
use_mirrored_aggregates	boolean	Specifies whether mirrored aggregates are selected when provisioning the volume to host the namespace. Only mirrored aggregates are used if this parameter is set to <i>true</i> and only unmirrored aggregates are used if this parameter is set to <i>false</i> . The default value is <i>true</i> for a MetroCluster configuration and is <i>false</i> for a non-MetroCluster configuration.

guarantee

Properties that request and report the space guarantee for the NVMe namespace.

Name	Type	Description
requested	boolean	<p>The requested space reservation policy for the NVMe namespace. If <i>true</i>, a space reservation is requested for the namespace; if <i>false</i>, the namespace is thin provisioned. Guaranteeing a space reservation request for a namespace requires that the volume in which the namespace resides also be space reserved and that the fractional reserve for the volume be 100%.</p> <p>The space reservation policy for an NVMe namespace is determined by ONTAP.</p> <ul style="list-style-type: none"><li>• readOnly: 1</li><li>• Introduced in: 9.6</li><li>• x-nullable: true</li></ul>
reserved	boolean	<p>Reports if the NVMe namespace is space guaranteed.</p> <p>This property is <i>true</i> if a space guarantee is requested and the containing volume and aggregate support the request. This property is <i>false</i> if a space guarantee is not requested or if a space guarantee is requested and either the containing volume and aggregate do not support the request.</p>

space

The storage space related properties of the NVMe namespace.

Name	Type	Description
block_size	integer	<p>The size of blocks in the namespace in bytes. The default for namespaces with an <code>os_type</code> of <i>vmware</i> is 512. All other namespaces default to 4096.</p> <p>Valid in POST when creating an NVMe namespace that is not a clone of another. Disallowed in POST when creating a namespace clone. Valid in POST.</p>
efficiency_ratio	number	<p>The storage efficiency ratio of the namespace without snapshots. (Logical Used / Used) This property is not available on the namespace object in the REST API and is not reported for GET requests. See the containing volume object for this information.</p>
guarantee	<a href="#">guarantee</a>	<p>Properties that request and report the space guarantee for the NVMe namespace.</p>
physical_used	integer	<p>The number of bytes consumed on the disk by the namespace, excluding snapshots. This property is not available on the namespace object in the REST API and is not reported for GET requests. See the containing volume object for this information.</p>
physical_used_by_snapshots	integer	<p>The number of bytes consumed on the disk by the namespace's snapshots. This property is not available on the namespace object in the REST API and is not reported for GET requests. See the containing volume object for this information.</p>

Name	Type	Description
size	integer	<p>The total provisioned size of the NVMe namespace. Valid in POST and PATCH. The NVMe namespace size can be increased but not be made smaller using the REST interface.</p> <p>The maximum and minimum sizes listed here are the absolute maximum and absolute minimum sizes in bytes. The maximum size is variable with respect to large NVMe namespace support in ONTAP. If large namespaces are supported, the maximum size is 128 TB (140737488355328 bytes) and if not supported, the maximum size is just under 16 TB (17557557870592 bytes). The minimum size supported is always 4096 bytes.</p> <p>For more information, see <i>Size properties</i> in the <i>docs</i> section of the ONTAP REST API documentation.</p> <ul style="list-style-type: none"> <li>• example: 1073741824</li> <li>• format: int64</li> <li>• Max value: 140737488355328</li> <li>• Min value: 4096</li> <li>• Introduced in: 9.6</li> <li>• x-nullable: true</li> </ul>



Name	Type	Description
used	integer	<p>The amount of space consumed by the main data stream of the NVMe namespace.</p> <p>This value is the total space consumed in the volume by the NVMe namespace, including filesystem overhead, but excluding prefix and suffix streams. Due to internal filesystem overhead and the many ways NVMe filesystems and applications utilize blocks within a namespace, this value does not necessarily reflect actual consumption/availability from the perspective of the filesystem or application. Without specific knowledge of how the namespace blocks are utilized outside of ONTAP, this property should not be used and an indicator for an out-of-space condition.</p> <p>For more information, see <i>Size properties</i> in the <i>docs</i> section of the ONTAP REST API documentation.</p> <ul style="list-style-type: none"> <li>• format: int64</li> <li>• readOnly: 1</li> <li>• Introduced in: 9.6</li> <li>• x-nullable: true</li> </ul>

#### iops\_raw

The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.

Name	Type	Description
other	integer	<p>Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.</p>

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### latency\_raw

The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### throughput\_raw

Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## statistics

These are raw performance numbers, such as IOPS latency and throughput. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.

Name	Type	Description
iops_raw	<a href="#">iops_raw</a>	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.
latency_raw	<a href="#">latency_raw</a>	The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput_raw	<a href="#">throughput_raw</a>	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.

Name	Type	Description
timestamp	string	The timestamp of the performance data.

status

Status information about the NVMe namespace.

Name	Type	Description
container_state	string	The state of the volume and aggregate that contain the NVMe namespace. Namespaces are only available when their containers are available.
mapped	boolean	Reports if the NVMe namespace is mapped to an NVMe subsystem.  There is an added computational cost to retrieving this property's value. It is not populated for a GET request unless it is explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more.
read_only	boolean	Reports if the NVMe namespace allows only read access.
state	string	The state of the NVMe namespace. Normal states for a namespace are <i>online</i> and <i>offline</i> . Other states indicate errors.

consistency\_group\_nvme\_host\_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	<p>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.</p> <p>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.</p>
group_size	string	<p>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.</p>
hash_function	string	<p>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.</p>

Name	Type	Description
host_secret_key	string	<p>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.</p> <p>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.</p>
mode	string	<p>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:</p> <ul style="list-style-type: none"> <li>• none: The host has neither the host nor controller secret configured, and no authentication is performed.</li> <li>• unidirectional: The host has a host secret configured. The controller will authenticate the host.</li> <li>• bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.</li> </ul>

tls

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

Name	Type	Description
configured_psk	string	<p>A user supplied pre-shared key (PSK) value in PSK Interchange Format. Optional in POST.</p> <p>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>.</p> <p>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.</p>

Name	Type	Description
key_type	string	<p>The method by which the TLS pre-shared key (PSK) is configured for the host. Optional in POST.</p> <p>The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• <code>none</code> - TLS is not configured for the host connection. No value is allowed for property <code>configured_psk</code>.</li> <li>• <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.</li> </ul> <p>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>.</p>

## hosts

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

Name	Type	Description
dh_hmac_chap	<a href="#">consistency_group_nvme_host_dh_hmac_chap</a>	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 io_queue count and I/O queue depth are being used.
tls	<a href="#">tls</a>	A container for the configuration for NVMe/TCP-TLS transport session for the host.

### subsystem

The NVMe subsystem to which the NVMe namespace is mapped.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
comment	string	A configurable comment for the NVMe subsystem. Optional in POST.
hosts	array[ <a href="#">hosts</a> ]	The NVMe hosts configured for access to the NVMe subsystem. Optional in POST.
name	string	The name of the NVMe subsystem. Once created, an NVMe subsystem cannot be renamed. Optional in POST.
os_type	string	The host operating system of the NVMe subsystem's hosts. Optional in POST.
uuid	string	The unique identifier of the NVMe subsystem.

### subsystem\_map

The NVMe subsystem with which the NVMe namespace is associated. A namespace can be mapped to zero (0) or one (1) subsystems.

There is an added computational cost to retrieving property values for `subsystem_map`. They are not populated for a GET request unless explicitly requested using the `fields` query parameter. See [Requesting specific fields](#) to learn more. These properties are supported only for GET.

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

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
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\_namespace

An NVMe namespace is a collection of addressable logical blocks presented to hosts connected to the storage virtual machine using the NVMe over Fabrics protocol.

An NVMe namespace must be mapped to an NVMe subsystem to grant access to the subsystem's hosts. Hosts can then access the NVMe namespace and perform I/O using the NVMe over Fabrics protocol.

See the NVMe namespace object model to learn more about each of the properties supported by the NVMe namespace REST API. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.

NVMe namespace names are paths of the form `"/vol/<volume>[/<qtree>]/<namespace>"` where the qtree name is optional.

An NVMe namespace is created to a specified size using thin or thick provisioning as determined by the volume on which it is created. An NVMe namespace can then be resized or cloned. An NVMe namespace cannot be renamed, or moved to a different volume. NVMe namespaces do not support the assignment of a QoS policy for performance management, but a QoS policy can be assigned to the volume containing the namespace.

`</namespace></qtree></volume>`

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
<code>auto_delete</code>	boolean	<p>This property marks the NVMe namespace for auto deletion when the volume containing the namespace runs out of space. This is most commonly set on namespace clones.</p> <p>When set to <i>true</i>, the NVMe namespace becomes eligible for automatic deletion when the volume runs out of space. Auto deletion only occurs when the volume containing the namespace is also configured for auto deletion and free space in the volume decreases below a particular threshold.</p> <p>This property is optional in POST and PATCH. The default value for a new NVMe namespace is <i>false</i>.</p> <p>There is an added computational cost to retrieving this property's value. It is not populated for a GET request unless it is explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more.</p>

Name	Type	Description
clone	<a href="#">clone</a>	<p>This sub-object is used in POST to create a new NVMe namespace as a clone of an existing namespace, or PATCH to overwrite an existing namespace as a clone of another. Setting a property in this sub-object indicates that a namespace clone is desired.</p> <p>When used in a PATCH, the patched NVMe namespace's data is over-written as a clone of the source and the following properties are preserved from the patched namespace unless otherwise specified as part of the PATCH: <code>auto_delete</code> (unless specified in the request), <code>subsystem_map</code>, <code>status.state</code>, and <code>uuid</code>.</p>
comment	string	A configurable comment available for use by the administrator. Valid in POST and PATCH.
consistency_group	<a href="#">consistency_group</a>	The namespace's consistency group. This property is populated for namespaces that are members of a consistency group. If the namespace is a member of a child consistency group, the parent consistency group is reported. A namespace's consistency group is the consistency group of its containing volume.
convert	<a href="#">convert</a>	This sub-object is used in POST to convert a valid in-place LUN to an NVMe namespace. Setting a property in this sub-object indicates that a conversion from the specified LUN to NVMe namespace is desired.
create_time	string	The time the NVMe namespace was created.

Name	Type	Description
enabled	boolean	The enabled state of the NVMe namespace. Certain error conditions cause the namespace to become disabled. If the namespace is disabled, check the <code>status.state</code> property to determine what error disabled the namespace. An NVMe namespace is enabled automatically when it is created.
location	<a href="#">location</a>	<p>The location of the NVMe namespace within the ONTAP cluster. NVMe namespaces do not support rename, or movement between volumes. Valid in POST.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.6</li> <li>• readCreate: 1</li> </ul>
metric	<a href="#">metric</a>	Performance numbers, such as IOPS latency and throughput
name	string	<p>The name of the NVMe namespace. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.</p> <p>NVMe namespace names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;namespace&gt;"</code> where the qtree name is optional.</p> <p>Renaming an NVMe namespace is not supported. Valid in POST.</p>
os_type	string	<p>The operating system type of the NVMe namespace.</p> <p>Required in POST when creating an NVMe namespace that is not a clone of another. Disallowed in POST when creating a namespace clone.</p>

Name	Type	Description
provisioning_options	<a href="#">provisioning_options</a>	Options that are applied to the operation.
space	<a href="#">space</a>	The storage space related properties of the NVMe namespace.
statistics	<a href="#">statistics</a>	These are raw performance numbers, such as IOPS latency and throughput. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.
status	<a href="#">status</a>	Status information about the NVMe namespace.
subsystem_map	<a href="#">subsystem_map</a>	<p>The NVMe subsystem with which the NVMe namespace is associated. A namespace can be mapped to zero (0) or one (1) subsystems.</p> <p>There is an added computational cost to retrieving property values for <code>subsystem_map</code>. They are not populated for a GET request unless explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more. These properties are supported only for GET.</p>
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.
uuid	string	The unique identifier of the NVMe namespace.

#### error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

#### returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

## Create an NVMe namespace

POST /storage/namespaces

**Introduced In:** 9.6

Creates an NVMe namespace.

### Required properties

- `svm.uuid` or `svm.name` - Existing SVM in which to create the NVMe namespace.
- `name`, `location.volume.name` or `location.volume.uuid` - Existing volume in which to create the NVMe namespace.
- `name` or `location.namespace` - Base name for the NVMe namespace.
- `os_type` - Operating system from which the NVMe namespace will be accessed. (Not used for clones, which are created based on the `os_type` of the source NVMe namespace.)
- `space.size` - Size for the NVMe namespace. (Not used for clones, which are created based on the size of the source NVMe namespace.)

### Default property values

If not specified in POST, the following default property values are assigned:

- `auto_delete` - *false*
- `space.block_size` - 4096 ( 512 when 'os\_type' is *vmware* )

### Related ONTAP commands

- `volume file clone autodelete`
- `volume file clone create`
- `vserver nvme namespace convert-from-lun`
- `vserver nvme namespace create`

POST is asynchronous when creating a new namespace. It is synchronous when converting a LUN to a namespace via the `convert` property.

## Learn more

- [DOC /storage/namespaces](#)

## Parameters

Name	Type	In	Required	Description
return_timeout	integer	query	False	<p>The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202.</p> <ul style="list-style-type: none"><li>• Default value: 1</li><li>• Max value: 120</li><li>• Min value: 0</li></ul>
return_records	boolean	query	False	<p>The default is false. If set to true, the records are returned.</p> <ul style="list-style-type: none"><li>• Default value:</li></ul>

## Request Body



Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
<code>auto_delete</code>	boolean	<p>This property marks the NVMe namespace for auto deletion when the volume containing the namespace runs out of space. This is most commonly set on namespace clones.</p> <p>When set to <i>true</i>, the NVMe namespace becomes eligible for automatic deletion when the volume runs out of space. Auto deletion only occurs when the volume containing the namespace is also configured for auto deletion and free space in the volume decreases below a particular threshold.</p> <p>This property is optional in POST and PATCH. The default value for a new NVMe namespace is <i>false</i>.</p> <p>There is an added computational cost to retrieving this property's value. It is not populated for a GET request unless it is explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more.</p>
<code>clone</code>	<a href="#">clone</a>	<p>This sub-object is used in POST to create a new NVMe namespace as a clone of an existing namespace, or PATCH to overwrite an existing namespace as a clone of another. Setting a property in this sub-object indicates that a namespace clone is desired.</p> <p>When used in a PATCH, the patched NVMe namespace's data is over-written as a clone of the source and the following properties are preserved from the patched namespace unless otherwise specified as part of the PATCH: <code>auto_delete</code> (unless specified in the request), <code>subsystem_map</code>, <code>status.state</code>, and <code>uuid</code>.</p>

Name	Type	Description
comment	string	A configurable comment available for use by the administrator. Valid in POST and PATCH.
consistency_group	<a href="#">consistency_group</a>	The namespace's consistency group. This property is populated for namespaces that are members of a consistency group. If the namespace is a member of a child consistency group, the parent consistency group is reported. A namespace's consistency group is the consistency group of its containing volume.
convert	<a href="#">convert</a>	This sub-object is used in POST to convert a valid in-place LUN to an NVMe namespace. Setting a property in this sub-object indicates that a conversion from the specified LUN to NVMe namespace is desired.
create_time	string	The time the NVMe namespace was created.
enabled	boolean	The enabled state of the NVMe namespace. Certain error conditions cause the namespace to become disabled. If the namespace is disabled, check the <code>status.state</code> property to determine what error disabled the namespace. An NVMe namespace is enabled automatically when it is created.
location	<a href="#">location</a>	The location of the NVMe namespace within the ONTAP cluster. NVMe namespaces do not support rename, or movement between volumes. Valid in POST. <ul style="list-style-type: none"> <li>• Introduced in: 9.6</li> <li>• readCreate: 1</li> </ul>
metric	<a href="#">metric</a>	Performance numbers, such as IOPS latency and throughput

Name	Type	Description
name	string	<p>The name of the NVMe namespace. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.</p> <p>NVMe namespace names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;namespace&gt;"</code> where the qtree name is optional.</p> <p>Renaming an NVMe namespace is not supported. Valid in POST.</p>
os_type	string	<p>The operating system type of the NVMe namespace.</p> <p>Required in POST when creating an NVMe namespace that is not a clone of another. Disallowed in POST when creating a namespace clone.</p>
provisioning_options	<a href="#">provisioning_options</a>	Options that are applied to the operation.
space	<a href="#">space</a>	The storage space related properties of the NVMe namespace.
statistics	<a href="#">statistics</a>	These are raw performance numbers, such as IOPS latency and throughput. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.
status	<a href="#">status</a>	Status information about the NVMe namespace.

Name	Type	Description
subsystem_map	<a href="#">subsystem_map</a>	<p>The NVMe subsystem with which the NVMe namespace is associated. A namespace can be mapped to zero (0) or one (1) subsystems.</p> <p>There is an added computational cost to retrieving property values for <code>subsystem_map</code>. They are not populated for a GET request unless explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more. These properties are supported only for GET.</p>
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.
uuid	string	The unique identifier of the NVMe namespace.

## Example request

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "clone": {
    "source": {
      "name": "/vol/volume1/namespace1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
  },
  "comment": "string",
  "consistency_group": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "cgl",
    "uuid": "4abc2317-4332-9d37-93a0-20bd29c22df0"
  },
  "convert": {
    "lun": {
      "name": "/vol/volume1/lun1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
  },
  "create_time": "2018-06-04 15:00:00 -0400",
  "location": {
    "namespace": "namespace1",
    "node": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "node1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "qtree": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      }
    }
  }
}
```

```

    }
  },
  "id": 1,
  "name": "qt1"
},
"volume": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "volume1",
  "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
}
},
"metric": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "duration": "PT15S",
  "iops": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "latency": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "status": "ok",
  "throughput": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "timestamp": "2017-01-25 06:20:13 -0500"
},
"name": "/vol/volume1/mtree1/namespace1",
"os_type": "string",
"provisioning_options": {
  "qos_policy": {
    "_links": {
      "self": {

```

```

        "href": "/api/resourcelink"
    },
    },
    "name": "performance",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"snapshot_policy": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    },
    "name": "default",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"storage_service": {
    "name": "string"
},
"tiering": {
    "control": "string",
    "object_stores": [
        {
            "name": "string"
        }
    ],
    "policy": "string"
}
},
"space": {
    "block_size": 512,
    "efficiency_ratio": 2.5,
    "physical_used": 1073741824,
    "physical_used_by_snapshots": 1073741824,
    "size": 1073741824,
    "used": 0
},
"statistics": {
    "iops_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "latency_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    }
}

```

```

    },
    "status": "ok",
    "throughput_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
},
"status": {
    "container_state": "string",
    "state": "online"
},
"subsystem_map": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "anagrpId": "00103050h",
    "nsid": "00000001h",
    "subsystem": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "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"
                }
            }
        ]
    }
}

```



```

    }
  ],
  "name": "subsystem1",
  "os_type": "string",
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
},
"svm": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "svm1",
  "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
},
"uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}

```

## Response

Status: 202, Accepted

Name	Type	Description
job	<a href="#">job_link</a>	

## Example response

```

{
  "job": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "uuid": "string"
  }
}

```

## Headers

Name	Description	Type
Location	Useful for tracking the resource location	string

## Response

Status: 201, Created

## Error

Status: Default

## ONTAP Error Response Codes

Error Code	Description
917927	The specified volume was not found.
918236	The specified <code>location.volume.uuid</code> and <code>location.volume.name</code> do not refer to the same volume.
1254197	The LUN specified for conversion to a namespace is mapped.
1260121	Cloning a namespace to a volume different than the source volume is not supported.
1260136	The specified destination for a clone operation already exists as a LUN, namespace, or file.
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	No SVM was specified. Either <code>svm.name</code> or <code>svm.uuid</code> must be supplied.
5242927	The specified <code>qtree</code> was not found.
5242950	The specified <code>location.qtree.id</code> and <code>location.qtree.name</code> do not refer to the same <code>qtree</code> .
5374127	The specified namespace name is invalid.
5374140	LUN has a non-zero prefix and/or suffix size.
5374141	LUN is part of a SnapMirror active sync relationship.
5374156	A protocol endpoint LUN cannot be converted to an NVMe namespace.

Error Code	Description
5374157	LUN in an SVM with MetroCluster configured cannot be converted to an NVMe namespace.
5374158	LUN contains an operating system type that is not supported for NVMe namespace.
5374352	An invalid name was provided for the NVMe namespace.
5374858	The volume specified by <code>name</code> is not the same as that specified by <code>location.volume</code> .
5374860	The qtree specified by <code>name</code> is not the same as that specified by <code>location.qtree</code> .
5374861	The NVMe namespace base name specified by <code>name</code> is not the same as that specified by <code>location.name</code> .
5374862	No NVMe namespace path base name was provided for the namespace.
5374876	The LUN specified for conversion to a namespace was not found.
5376461	The specified namespace name is invalid.
5376462	The specified namespace name is too long.
5376463	The snapshot portion of the specified namespace name is too long.
5440509	No suitable storage can be found for the specified requirements.
13565952	The NVMe namespace clone request failed.
72089636	Creating NVMe namespaces with <code>os_type</code> AIX is not supported until the effective cluster version is 9.13.1 or later.
72089720	NVMe namespaces cannot be created in snapshots.
72089721	The volume specified is in a load sharing mirror relationship. Namespaces are not supported in load sharing mirrors.
72089722	A negative size was provided for the NVMe namespace.
72089723	The specified size is too small for the NVMe namespace.
72089724	The specified size is too large for the NVMe namespace.
72089725	A LUN or NVMe namespace already exists at the specified path.
72089727	NVMe namespaces cannot be created on an SVM root volume.

Error Code	Description
72089728	NVMe namespaces cannot be created on a FlexGroup volume.
72089732	An NVMe namespace name can only contain characters A-Z, a-z, 0-9, "-", ".", "_", "{" and "}".
72090005	The specified <code>clone.source.uuid</code> and <code>clone.source.name</code> do not refer to the same NVMe namespace.
72090006	The specified <code>clone.source</code> was not found.
72090007	The specified <code>clone.source</code> was not found.
72090009	An error occurred after successfully creating the NVMe namespace. Some properties were not set.
72090012	The property cannot be specified when creating an NVMe namespace clone. The <code>target</code> property of the error object identifies the property.
72090013	The property is required except when creating an NVMe namespace clone. The <code>target</code> property of the error object identifies the property.
72090014	No volume was specified for the NVMe namespace.
72090015	An error occurred after successfully creating the NVMe namespace preventing the retrieval of its properties.
72090033	The <code>clone.source.uuid</code> property is not supported when specifying a source NVMe namespace from a snapshot.
72090039	The property cannot be specified at the same time when creating an NVMe namespace as a clone. The <code>target</code> property of the error object identifies the other property given with clone.
72090040	The property cannot be specified when converting a LUN into an NVMe namespace. The <code>target</code> property of the error object identifies the property.

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

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	

source

The source NVMe namespace for a namespace clone operation. This can be specified using property `clone.source.uuid` or `clone.source.name`. If both properties are supplied, they must refer to the same namespace.

Valid in POST to create a new NVMe namespace as a clone of the source.

Valid in PATCH to overwrite an existing NVMe namespace's data as a clone of another.

Name	Type	Description
name	string	<p>The name of the clone source NVMe namespace. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.</p> <p>NVMe namespace names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;namespace&gt;"</code> where the qtree name is optional.</p> <p>Valid in POST and PATCH.</p>
uuid	string	<p>The unique identifier of the clone source NVMe namespace. Valid in POST and PATCH.</p>

clone

This sub-object is used in POST to create a new NVMe namespace as a clone of an existing namespace, or PATCH to overwrite an existing namespace as a clone of another. Setting a property in this sub-object indicates that a namespace clone is desired.

When used in a PATCH, the patched NVMe namespace's data is over-written as a clone of the source and the following properties are preserved from the patched namespace unless otherwise specified as

part of the PATCH: `auto_delete` (unless specified in the request), `subsystem_map`, `status.state`, and `uuid`.

Name	Type	Description
source	<a href="#">source</a>	<p>The source NVMe namespace for a namespace clone operation. This can be specified using property <code>clone.source.uuid</code> or <code>clone.source.name</code>. If both properties are supplied, they must refer to the same namespace.</p> <p>Valid in POST to create a new NVMe namespace as a clone of the source.</p> <p>Valid in PATCH to overwrite an existing NVMe namespace's data as a clone of another.</p>

#### consistency\_group

The namespace's consistency group. This property is populated for namespaces that are members of a consistency group. If the namespace is a member of a child consistency group, the parent consistency group is reported. A namespace's consistency group is the consistency group of its containing volume.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
name	string	The name of the consistency group.
uuid	string	The unique identifier of the consistency group.

#### lun

The source LUN for convert operation. This can be specified using property `convert.lun.uuid` or `convert.lun.name`. If both properties are supplied, they must refer to the same LUN.

Valid in POST. A convert request from LUN to NVMe namespace cannot be combined with setting any other namespace properties. All other properties of the converted NVMe namespace come from the source LUN.

Name	Type	Description
name	string	The name of the source LUN. Valid in POST. A LUN is located within a volume. Optionally, it can be located within a qtree in a volume.  LUN names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;lun&gt;"</code> where the qtree name is optional.
uuid	string	The unique identifier of the source LUN. Valid in POST.

## convert

This sub-object is used in POST to convert a valid in-place LUN to an NVMe namespace. Setting a property in this sub-object indicates that a conversion from the specified LUN to NVMe namespace is desired.

Name	Type	Description
lun	<a href="#">lun</a>	The source LUN for convert operation. This can be specified using property <code>convert.lun.uuid</code> or <code>convert.lun.name</code> . If both properties are supplied, they must refer to the same LUN.  Valid in POST. A convert request from LUN to NVMe namespace cannot be combined with setting any other namespace properties. All other properties of the converted NVMe namespace come from the source LUN.

## node

The cluster node that hosts the NVMe namespace.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
name	string	
uuid	string	

## qtree



The qtree in which the NVMe namespace is optionally located. Valid in POST.

If properties `name` and `location.qtree.name` and/or `location.qtree.uuid` are specified in the same request, they must refer to the same qtree.

NVMe namespaces do not support rename.

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
<code>id</code>	integer	The identifier for the qtree, unique within the qtree's volume.
<code>name</code>	string	The name of the qtree.

volume

The volume in which the NVMe namespace is located. Valid in POST.

If properties `name` and `location.volume.name` and/or `location.volume.uuid` are specified in the same request, they must refer to the same volume.

NVMe namespaces do not support movement between volumes.

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
<code>name</code>	string	The name of the volume. This field cannot be specified in a PATCH method.
<code>uuid</code>	string	Unique identifier for the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move. <ul style="list-style-type: none"><li>• example: 028baa66-41bd-11e9-81d5-00a0986138f7</li><li>• Introduced in: 9.6</li><li>• x-nullable: true</li></ul>

location

The location of the NVMe namespace within the ONTAP cluster. NVMe namespaces do not support rename, or movement between volumes. Valid in POST.

Name	Type	Description
namespace	string	<p>The base name component of the NVMe namespace. Valid in POST.</p> <p>If properties <code>name</code> and <code>location.namespace</code> are specified in the same request, they must refer to the base name.</p> <p>NVMe namespaces do not support rename.</p>
node	node	<p>The cluster node that hosts the NVMe namespace.</p>
qtree	qtree	<p>The qtree in which the NVMe namespace is optionally located. Valid in POST.</p> <p>If properties <code>name</code> and <code>location.qtree.name</code> and/or <code>location.qtree.uuid</code> are specified in the same request, they must refer to the same qtree.</p> <p>NVMe namespaces do not support rename.</p>
volume	volume	<p>The volume in which the NVMe namespace is located. Valid in POST.</p> <p>If properties <code>name</code> and <code>location.volume.name</code> and/or <code>location.volume.uuid</code> are specified in the same request, they must refer to the same volume.</p> <p>NVMe namespaces do not support movement between volumes.</p>

iops

The rate of I/O operations observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### latency

The round trip latency in microseconds observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### throughput

The rate of throughput bytes per second observed at the storage object.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.

Name	Type	Description
write	integer	Performance metric for write I/O operations.

metric

Performance numbers, such as IOPS latency and throughput

Name	Type	Description
_links	<a href="#">_links</a>	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
iops	<a href="#">iops</a>	The rate of I/O operations observed at the storage object.
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.

Name	Type	Description
throughput	<a href="#">throughput</a>	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

`self_link`

Name	Type	Description
self	<a href="#">href</a>	

`qos_policy`

The QoS policy for the volume provisioned to host the namespace. This property is only supported when the request provisions a new volume. If no `qos_policy` is provided, a policy is be set based on the `storage_service.name`, which defaults to the most performant service available.

Name	Type	Description
<code>_links</code>	<a href="#">self_link</a>	
name	string	The QoS policy group name. This is mutually exclusive with UUID and other QoS attributes during POST and PATCH.
uuid	string	The QoS policy group UUID. This is mutually exclusive with name and other QoS attributes during POST and PATCH.

`snapshot_policy_reference`

This is a reference to the snapshot policy.

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
name	string	
uuid	string	

`storage_service`

Determines the placement of the namespace based on the value specified. Valid in POST.

Name	Type	Description
name	string	Storage service name. If not specified, the default value is the most performant for the platform.

#### object\_stores

Name	Type	Description
name	string	The name of the object store to use. Used for placement.

#### tiering

The tiering placement and policy definitions for the volume provisioned to host the namespace. This property is only supported when the request provisions a new volume.

Name	Type	Description
control	string	Storage tiering placement rules for the object.
object_stores	array[ <a href="#">object_stores</a> ]	Object stores to use. Used for placement.

Name	Type	Description
policy	string	<p>Policy that determines whether the user data blocks of a volume in a FabricPool will be tiered to the cloud store when they become cold.</p> <p>FabricPool combines flash (performance tier) with a cloud store into a single aggregate. Temperature of a volume block increases if it is accessed frequently and decreases when it is not. Valid in POST or PATCH.</p> <p>all &amp;dash; Allows tiering of both snapshots and active file system user data to the cloud store as soon as possible by ignoring the temperature on the volume blocks.</p> <p>auto &amp;dash; Allows tiering of both snapshot and active file system user data to the cloud store</p> <p>none &amp;dash; Volume blocks are not be tiered to the cloud store.</p> <p>snapshot_only &amp;dash; Allows tiering of only the volume snapshots not associated with the active file system.</p> <p>The default tiering policy is "snapshot-only" for a FlexVol volume and "none" for a FlexGroup volume. The default minimum cooling period for the "snapshot-only" tiering policy is 2 days and for the "auto" tiering policy it is 31 days.</p>

provisioning\_options

Options that are applied to the operation.





Name	Type	Description
auto	boolean	<p>If the volume specified in the request does not exist, automatically provision one of appropriate size. If the volume does exist, resize it to accommodate the new namespace.</p> <p>This property is only supported on Unified ONTAP.</p> <p>The following behavior is different from a traditional POST request:</p> <ul style="list-style-type: none"> <li>• The operation is asynchronous.</li> <li>• The <code>qos_policy</code> property is supported and is applied to the provisioned volume. A default QoS policy is applied to the volume if one is not provided.</li> <li>• The <code>provisioning_options.count</code> property is supported, provisioning <i>count</i> namespaces on the volume using the specified properties.</li> <li>• The <code>subsystem_map</code> property is supported. If the specified subsystem does not exist, it is created. The namespace is mapped to this subsystem. If a subsystem is provisioned in this way, it is deleted after it is no longer mapped to any namespaces.</li> <li>• The <code>clone</code> and <code>convert</code> properties are not supported.</li> <li>• When performing <code>records</code> based operations, specifying this property in the query applies to the entire operation. Specifying it for an individual record within the request applies to only that record.</li> <li>• Many other <code>provisioning_options</code> properties are supported to control the placement of the namespace and the properties of the volume</li> </ul>

Name	Type	Description
count	integer	The number of namespaces to provision with these properties. Only POST requests based on <code>space.size</code> are supported. When provided, the name is considered a prefix, and a suffix of the form <code>_&lt;N&gt;</code> is generated where N is the next available numeric index, starting with 1.
qos_policy	<a href="#">qos_policy</a>	The QoS policy for the volume provisioned to host the namespace. This property is only supported when the request provisions a new volume. If no <code>qos_policy</code> is provided, a policy is be set based on the <code>storage_service.name</code> , which defaults to the most performant service available.
snapshot_policy	<a href="#">snapshot_policy_reference</a>	This is a reference to the snapshot policy.
storage_service	<a href="#">storage_service</a>	Determines the placement of the namespace based on the value specified. Valid in POST.
tiering	<a href="#">tiering</a>	The tiering placement and policy definitions for the volume provisioned to host the namespace. This property is only supported when the request provisions a new volume.
use_mirrored_aggregates	boolean	Specifies whether mirrored aggregates are selected when provisioning the volume to host the namespace. Only mirrored aggregates are used if this parameter is set to <i>true</i> and only unmirrored aggregates are used if this parameter is set to <i>false</i> . The default value is <i>true</i> for a MetroCluster configuration and is <i>false</i> for a non-MetroCluster configuration.

guarantee

Properties that request and report the space guarantee for the NVMe namespace.

Name	Type	Description
requested	boolean	<p>The requested space reservation policy for the NVMe namespace. If <i>true</i>, a space reservation is requested for the namespace; if <i>false</i>, the namespace is thin provisioned. Guaranteeing a space reservation request for a namespace requires that the volume in which the namespace resides also be space reserved and that the fractional reserve for the volume be 100%.</p> <p>The space reservation policy for an NVMe namespace is determined by ONTAP.</p> <ul style="list-style-type: none"><li>• readOnly: 1</li><li>• Introduced in: 9.6</li><li>• x-nullable: true</li></ul>
reserved	boolean	<p>Reports if the NVMe namespace is space guaranteed.</p> <p>This property is <i>true</i> if a space guarantee is requested and the containing volume and aggregate support the request. This property is <i>false</i> if a space guarantee is not requested or if a space guarantee is requested and either the containing volume and aggregate do not support the request.</p>

space

The storage space related properties of the NVMe namespace.

Name	Type	Description
block_size	integer	<p>The size of blocks in the namespace in bytes. The default for namespaces with an <code>os_type</code> of <i>vmware</i> is 512. All other namespaces default to 4096.</p> <p>Valid in POST when creating an NVMe namespace that is not a clone of another. Disallowed in POST when creating a namespace clone. Valid in POST.</p>
efficiency_ratio	number	<p>The storage efficiency ratio of the namespace without snapshots. (Logical Used / Used) This property is not available on the namespace object in the REST API and is not reported for GET requests. See the containing volume object for this information.</p>
guarantee	<a href="#">guarantee</a>	<p>Properties that request and report the space guarantee for the NVMe namespace.</p>
physical_used	integer	<p>The number of bytes consumed on the disk by the namespace, excluding snapshots. This property is not available on the namespace object in the REST API and is not reported for GET requests. See the containing volume object for this information.</p>
physical_used_by_snapshots	integer	<p>The number of bytes consumed on the disk by the namespace's snapshots. This property is not available on the namespace object in the REST API and is not reported for GET requests. See the containing volume object for this information.</p>

Name	Type	Description
size	integer	<p>The total provisioned size of the NVMe namespace. Valid in POST and PATCH. The NVMe namespace size can be increased but not be made smaller using the REST interface.</p> <p>The maximum and minimum sizes listed here are the absolute maximum and absolute minimum sizes in bytes. The maximum size is variable with respect to large NVMe namespace support in ONTAP. If large namespaces are supported, the maximum size is 128 TB (140737488355328 bytes) and if not supported, the maximum size is just under 16 TB (17557557870592 bytes). The minimum size supported is always 4096 bytes.</p> <p>For more information, see <i>Size properties</i> in the <i>docs</i> section of the ONTAP REST API documentation.</p> <ul style="list-style-type: none"> <li>• example: 1073741824</li> <li>• format: int64</li> <li>• Max value: 140737488355328</li> <li>• Min value: 4096</li> <li>• Introduced in: 9.6</li> <li>• x-nullable: true</li> </ul>

Name	Type	Description
used	integer	<p>The amount of space consumed by the main data stream of the NVMe namespace.</p> <p>This value is the total space consumed in the volume by the NVMe namespace, including filesystem overhead, but excluding prefix and suffix streams. Due to internal filesystem overhead and the many ways NVMe filesystems and applications utilize blocks within a namespace, this value does not necessarily reflect actual consumption/availability from the perspective of the filesystem or application. Without specific knowledge of how the namespace blocks are utilized outside of ONTAP, this property should not be used and an indicator for an out-of-space condition.</p> <p>For more information, see <i>Size properties</i> in the <i>docs</i> section of the ONTAP REST API documentation.</p> <ul style="list-style-type: none"> <li>• format: int64</li> <li>• readOnly: 1</li> <li>• Introduced in: 9.6</li> <li>• x-nullable: true</li> </ul>

#### iops\_raw

The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.

Name	Type	Description
other	integer	<p>Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.</p>

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### latency\_raw

The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### throughput\_raw

Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## statistics

These are raw performance numbers, such as IOPS latency and throughput. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.

Name	Type	Description
iops_raw	<a href="#">iops_raw</a>	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.
latency_raw	<a href="#">latency_raw</a>	The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput_raw	<a href="#">throughput_raw</a>	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.



Name	Type	Description
timestamp	string	The timestamp of the performance data.

status

Status information about the NVMe namespace.

Name	Type	Description
container_state	string	The state of the volume and aggregate that contain the NVMe namespace. Namespaces are only available when their containers are available.
mapped	boolean	Reports if the NVMe namespace is mapped to an NVMe subsystem.  There is an added computational cost to retrieving this property's value. It is not populated for a GET request unless it is explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more.
read_only	boolean	Reports if the NVMe namespace allows only read access.
state	string	The state of the NVMe namespace. Normal states for a namespace are <i>online</i> and <i>offline</i> . Other states indicate errors.

consistency\_group\_nvme\_host\_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	<p>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.</p> <p>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.</p>
group_size	string	<p>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.</p>
hash_function	string	<p>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.</p>

Name	Type	Description
host_secret_key	string	<p>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.</p> <p>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.</p>
mode	string	<p>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:</p> <ul style="list-style-type: none"> <li>• none: The host has neither the host nor controller secret configured, and no authentication is performed.</li> <li>• unidirectional: The host has a host secret configured. The controller will authenticate the host.</li> <li>• bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.</li> </ul>

tls

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

Name	Type	Description
configured_psk	string	<p>A user supplied pre-shared key (PSK) value in PSK Interchange Format. Optional in POST.</p> <p>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>.</p> <p>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.</p>

Name	Type	Description
key_type	string	<p>The method by which the TLS pre-shared key (PSK) is configured for the host. Optional in POST.</p> <p>The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• <code>none</code> - TLS is not configured for the host connection. No value is allowed for property <code>configured_psk</code>.</li> <li>• <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.</li> </ul> <p>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>.</p>

## hosts

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

Name	Type	Description
dh_hmac_chap	<a href="#">consistency_group_nvme_host_dh_hmac_chap</a>	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 io_queue count and I/O queue depth are being used.
tls	<a href="#">tls</a>	A container for the configuration for NVMe/TCP-TLS transport session for the host.

### subsystem

The NVMe subsystem to which the NVMe namespace is mapped.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
comment	string	A configurable comment for the NVMe subsystem. Optional in POST.
hosts	array[ <a href="#">hosts</a> ]	The NVMe hosts configured for access to the NVMe subsystem. Optional in POST.
name	string	The name of the NVMe subsystem. Once created, an NVMe subsystem cannot be renamed. Optional in POST.
os_type	string	The host operating system of the NVMe subsystem's hosts. Optional in POST.
uuid	string	The unique identifier of the NVMe subsystem.

### subsystem\_map

The NVMe subsystem with which the NVMe namespace is associated. A namespace can be mapped to zero (0) or one (1) subsystems.

There is an added computational cost to retrieving property values for `subsystem_map`. They are not populated for a GET request unless explicitly requested using the `fields` query parameter. See [Requesting specific fields](#) to learn more. These properties are supported only for GET.

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

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
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\_namespace

An NVMe namespace is a collection of addressable logical blocks presented to hosts connected to the storage virtual machine using the NVMe over Fabrics protocol.

An NVMe namespace must be mapped to an NVMe subsystem to grant access to the subsystem's hosts. Hosts can then access the NVMe namespace and perform I/O using the NVMe over Fabrics protocol.

See the NVMe namespace object model to learn more about each of the properties supported by the NVMe namespace REST API. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.

NVMe namespace names are paths of the form `"/vol/<volume>[/<qtree>]/<namespace>"` where the qtree name is optional.

An NVMe namespace is created to a specified size using thin or thick provisioning as determined by the volume on which it is created. An NVMe namespace can then be resized or cloned. An NVMe namespace cannot be renamed, or moved to a different volume. NVMe namespaces do not support the assignment of a QoS policy for performance management, but a QoS policy can be assigned to the volume containing the namespace.

`</namespace></qtree></volume>`

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
<code>auto_delete</code>	boolean	<p>This property marks the NVMe namespace for auto deletion when the volume containing the namespace runs out of space. This is most commonly set on namespace clones.</p> <p>When set to <i>true</i>, the NVMe namespace becomes eligible for automatic deletion when the volume runs out of space. Auto deletion only occurs when the volume containing the namespace is also configured for auto deletion and free space in the volume decreases below a particular threshold.</p> <p>This property is optional in POST and PATCH. The default value for a new NVMe namespace is <i>false</i>.</p> <p>There is an added computational cost to retrieving this property's value. It is not populated for a GET request unless it is explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more.</p>



Name	Type	Description
clone	<a href="#">clone</a>	<p>This sub-object is used in POST to create a new NVMe namespace as a clone of an existing namespace, or PATCH to overwrite an existing namespace as a clone of another. Setting a property in this sub-object indicates that a namespace clone is desired.</p> <p>When used in a PATCH, the patched NVMe namespace's data is over-written as a clone of the source and the following properties are preserved from the patched namespace unless otherwise specified as part of the PATCH: <code>auto_delete</code> (unless specified in the request), <code>subsystem_map</code>, <code>status.state</code>, and <code>uuid</code>.</p>
comment	string	A configurable comment available for use by the administrator. Valid in POST and PATCH.
consistency_group	<a href="#">consistency_group</a>	The namespace's consistency group. This property is populated for namespaces that are members of a consistency group. If the namespace is a member of a child consistency group, the parent consistency group is reported. A namespace's consistency group is the consistency group of its containing volume.
convert	<a href="#">convert</a>	This sub-object is used in POST to convert a valid in-place LUN to an NVMe namespace. Setting a property in this sub-object indicates that a conversion from the specified LUN to NVMe namespace is desired.
create_time	string	The time the NVMe namespace was created.

Name	Type	Description
enabled	boolean	The enabled state of the NVMe namespace. Certain error conditions cause the namespace to become disabled. If the namespace is disabled, check the <code>status.state</code> property to determine what error disabled the namespace. An NVMe namespace is enabled automatically when it is created.
location	<a href="#">location</a>	<p>The location of the NVMe namespace within the ONTAP cluster. NVMe namespaces do not support rename, or movement between volumes. Valid in POST.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.6</li> <li>• readCreate: 1</li> </ul>
metric	<a href="#">metric</a>	Performance numbers, such as IOPS latency and throughput
name	string	<p>The name of the NVMe namespace. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.</p> <p>NVMe namespace names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;namespace&gt;"</code> where the qtree name is optional.</p> <p>Renaming an NVMe namespace is not supported. Valid in POST.</p>
os_type	string	<p>The operating system type of the NVMe namespace.</p> <p>Required in POST when creating an NVMe namespace that is not a clone of another. Disallowed in POST when creating a namespace clone.</p>

Name	Type	Description
provisioning_options	<a href="#">provisioning_options</a>	Options that are applied to the operation.
space	<a href="#">space</a>	The storage space related properties of the NVMe namespace.
statistics	<a href="#">statistics</a>	These are raw performance numbers, such as IOPS latency and throughput. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.
status	<a href="#">status</a>	Status information about the NVMe namespace.
subsystem_map	<a href="#">subsystem_map</a>	<p>The NVMe subsystem with which the NVMe namespace is associated. A namespace can be mapped to zero (0) or one (1) subsystems.</p> <p>There is an added computational cost to retrieving property values for <code>subsystem_map</code>. They are not populated for a GET request unless explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more. These properties are supported only for GET.</p>
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.
uuid	string	The unique identifier of the NVMe namespace.

#### job\_link

Name	Type	Description
_links	<a href="#">_links</a>	
uuid	string	The UUID of the asynchronous job that is triggered by a POST, PATCH, or DELETE operation.

#### 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 historical performance metrics for an NVMe namespace

GET /storage/namespaces/{nvme\_namespace.uuid}/metrics

**Introduced In:** 9.14

Retrieves historical space and performance metrics for an NVMe namespace.

### Related ONTAP commands

- `statistics namespace show`

### Parameters

Name	Type	In	Required	Description
duration	string	query	False	Filter by duration
latency.write	integer	query	False	Filter by latency.write
latency.total	integer	query	False	Filter by latency.total
latency.read	integer	query	False	Filter by latency.read

Name	Type	In	Required	Description
latency.other	integer	query	False	Filter by latency.other
timestamp	string	query	False	Filter by timestamp
iops.write	integer	query	False	Filter by iops.write
iops.total	integer	query	False	Filter by iops.total
iops.read	integer	query	False	Filter by iops.read
iops.other	integer	query	False	Filter by iops.other
uuid	string	query	False	Filter by uuid
status	string	query	False	Filter by status
throughput.total	integer	query	False	Filter by throughput.total
throughput.write	integer	query	False	Filter by throughput.write
throughput.read	integer	query	False	Filter by throughput.read
nvme_namespace.uuid	string	path	True	The unique identifier of the NVMe namespace.

Name	Type	In	Required	Description
interval	string	query	False	<p>The time range for the data. Examples can be 1h, 1d, 1m, 1w, 1y. The period for each time range is as follows:</p> <ul style="list-style-type: none"> <li>• 1h: Metrics over the most recent hour sampled over 15 seconds.</li> <li>• 1d: Metrics over the most recent day sampled over 5 minutes.</li> <li>• 1w: Metrics over the most recent week sampled over 30 minutes.</li> <li>• 1m: Metrics over the most recent month sampled over 2 hours.</li> <li>• 1y: Metrics over the most recent year sampled over a day.</li> <li>• Default value: 1</li> <li>• enum: ["1h", "1d", "1w", "1m", "1y"]</li> </ul>

Name	Type	In	Required	Description
return_timeout	integer	query	False	<p>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.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> <li>• Max value: 120</li> <li>• Min value: 0</li> </ul>
fields	array[string]	query	False	Specify the fields to return.
max_records	integer	query	False	Limit the number of records returned.
order_by	array[string]	query	False	Order results by specified fields and optional [asc
desc] direction. Default direction is 'asc' for ascending.	return_records	boolean	query	False

## Response

Status: 200, Ok

Name	Type	Description
_links	<a href="#">_links</a>	
num_records	integer	Number of records
records	array[ <a href="#">records</a> ]	

## Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": 1,
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "duration": "PT15S",
      "iops": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "latency": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "status": "ok",
      "throughput": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "timestamp": "2017-01-25 06:20:13 -0500",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
  ]
}
```



## Error

Status: Default

### ONTAP Error Response Codes

Error Code	Description
8585947	No metrics are available for the requested object.
8586225	An unexpected error occurred retrieving metrics for the requested object.

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

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	
self	<a href="#">href</a>	

\_links

Name	Type	Description
self	<a href="#">href</a>	

iops

The rate of I/O operations observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

latency

The round trip latency in microseconds observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## throughput

The rate of throughput bytes per second observed at the storage object.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## records

Performance numbers, such as IOPS latency and throughput, for SVM protocols.

Name	Type	Description
_links	<a href="#">_links</a>	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
iops	<a href="#">iops</a>	The rate of I/O operations observed at the storage object.

Name	Type	Description
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput	<a href="#">throughput</a>	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.
uuid	string	The unique identifier of the NVMe namespace.

#### error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

#### returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

## Retrieve historical performance metrics for an NVMe namespace for a specific time

GET /storage/namespaces/{nvme\_namespace.uuid}/metrics/{timestamp}

**Introduced In:** 9.14

Retrieves historical space and performance metrics for an NVMe namespace for a specific time.

### Related ONTAP commands

- `statistics namespace show`

### Parameters

Name	Type	In	Required	Description
nvme_namespace.uuid	string	path	True	The unique identifier of the NVMe namespace.
timestamp	string	path	True	The timestamp of the performance data. <ul style="list-style-type: none"> <li>• format: date-time</li> </ul>
fields	array[string]	query	False	Specify the fields to return.

### Response

Status: 200, Ok

Name	Type	Description
_links	<a href="#">_links</a>	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
iops	<a href="#">iops</a>	The rate of I/O operations observed at the storage object.
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput	<a href="#">throughput</a>	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.
uuid	string	The unique identifier of the NVMe namespace.

## Example response

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "duration": "PT15S",
  "iops": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "latency": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "status": "ok",
  "throughput": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "timestamp": "2017-01-25 06:20:13 -0500",
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
```

## Error

Status: Default

## ONTAP Error Response Codes

Error Code	Description
8585947	No metrics are available for the requested object.
8586225	An unexpected error occurred retrieving metrics for the requested object.

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

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	

iops

The rate of I/O operations observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

latency

The round trip latency in microseconds observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.

Name	Type	Description
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### throughput

The rate of throughput bytes per second observed at the storage object.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

#### returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

## Delete an NVMe namespace

```
DELETE /storage/namespaces/{uuid}
```

**Introduced In:** 9.6

Deletes an NVMe namespace.

### Related ONTAP commands

- `vserver nvme namespace delete`

### Learn more

- [DOC /storage/namespaces](#)

### Parameters

Name	Type	In	Required	Description
uuid	string	path	True	The unique identifier of the NVMe namespace to delete.
allow_delete_while_mapped	boolean	query	False	<p>Allows deletion of a mapped NVMe namespace. A mapped NVMe namespace might be in use. Deleting a mapped namespace also deletes the namespace map and makes the data no longer available, possibly causing a disruption in the availability of data.</p> <p><b>This parameter should be used with caution.</b></p> <ul style="list-style-type: none"><li>• Default value:</li></ul>

Name	Type	In	Required	Description
return_timeout	integer	query	False	<p>The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> <li>• Max value: 120</li> <li>• Min value: 0</li> </ul>

## Response

Status: 200, Ok

## Response

Status: 202, Accepted

## Error

Status: Default

## ONTAP Error Response Codes

Error Code	Description
4	The specified namespace was not found.
72089796	The namespace must be unmapped before deletion.
72090016	The namespace's aggregate is offline. The aggregate must be online to modify or remove the namespace.
72090017	The namespace's volume is offline. The volume must be online to modify or remove the namespace.

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

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

## Retrieve an NVMe namespace

GET /storage/namespaces/{uuid}

**Introduced In:** 9.6

Retrieves an NVMe namespace.

### 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.

- `auto_delete`
- `space.physical_used`
- `space.physical_used_by_snapshots`
- `space.efficiency_ratio`
- `subsystem_map.*`
- `status.mapped`
- `statistics.*`
- `metric.*`

## Related ONTAP commands

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

## Learn more

- [DOC /storage/namespaces](#)

## Parameters

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

## Response

Status: 200, Ok

Name	Type	Description
_links	<a href="#">_links</a>	

Name	Type	Description
auto_delete	boolean	<p>This property marks the NVMe namespace for auto deletion when the volume containing the namespace runs out of space. This is most commonly set on namespace clones.</p> <p>When set to <i>true</i>, the NVMe namespace becomes eligible for automatic deletion when the volume runs out of space. Auto deletion only occurs when the volume containing the namespace is also configured for auto deletion and free space in the volume decreases below a particular threshold.</p> <p>This property is optional in POST and PATCH. The default value for a new NVMe namespace is <i>false</i>.</p> <p>There is an added computational cost to retrieving this property's value. It is not populated for a GET request unless it is explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more.</p>
clone	<a href="#">clone</a>	<p>This sub-object is used in POST to create a new NVMe namespace as a clone of an existing namespace, or PATCH to overwrite an existing namespace as a clone of another. Setting a property in this sub-object indicates that a namespace clone is desired.</p> <p>When used in a PATCH, the patched NVMe namespace's data is over-written as a clone of the source and the following properties are preserved from the patched namespace unless otherwise specified as part of the PATCH: <code>auto_delete</code> (unless specified in the request), <code>subsystem_map</code>, <code>status.state</code>, and <code>uuid</code>.</p>



Name	Type	Description
comment	string	A configurable comment available for use by the administrator. Valid in POST and PATCH.
consistency_group	<a href="#">consistency_group</a>	The namespace's consistency group. This property is populated for namespaces that are members of a consistency group. If the namespace is a member of a child consistency group, the parent consistency group is reported. A namespace's consistency group is the consistency group of its containing volume.
convert	<a href="#">convert</a>	This sub-object is used in POST to convert a valid in-place LUN to an NVMe namespace. Setting a property in this sub-object indicates that a conversion from the specified LUN to NVMe namespace is desired.
create_time	string	The time the NVMe namespace was created.
enabled	boolean	The enabled state of the NVMe namespace. Certain error conditions cause the namespace to become disabled. If the namespace is disabled, check the <code>status.state</code> property to determine what error disabled the namespace. An NVMe namespace is enabled automatically when it is created.
location	<a href="#">location</a>	The location of the NVMe namespace within the ONTAP cluster. NVMe namespaces do not support rename, or movement between volumes. Valid in POST. <ul style="list-style-type: none"> <li>• Introduced in: 9.6</li> <li>• readCreate: 1</li> </ul>
metric	<a href="#">metric</a>	Performance numbers, such as IOPS latency and throughput

Name	Type	Description
name	string	<p>The name of the NVMe namespace. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.</p> <p>NVMe namespace names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;namespace&gt;"</code> where the qtree name is optional.</p> <p>Renaming an NVMe namespace is not supported. Valid in POST.</p>
os_type	string	<p>The operating system type of the NVMe namespace.</p> <p>Required in POST when creating an NVMe namespace that is not a clone of another. Disallowed in POST when creating a namespace clone.</p>
provisioning_options	<a href="#">provisioning_options</a>	Options that are applied to the operation.
space	<a href="#">space</a>	The storage space related properties of the NVMe namespace.
statistics	<a href="#">statistics</a>	These are raw performance numbers, such as IOPS latency and throughput. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.
status	<a href="#">status</a>	Status information about the NVMe namespace.

Name	Type	Description
subsystem_map	<a href="#">subsystem_map</a>	<p>The NVMe subsystem with which the NVMe namespace is associated. A namespace can be mapped to zero (0) or one (1) subsystems.</p> <p>There is an added computational cost to retrieving property values for <code>subsystem_map</code>. They are not populated for a GET request unless explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more. These properties are supported only for GET.</p>
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.
uuid	string	The unique identifier of the NVMe namespace.

## Example response

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "clone": {
    "source": {
      "name": "/vol/volume1/namespace1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
  },
  "comment": "string",
  "consistency_group": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "cgl",
    "uuid": "4abc2317-4332-9d37-93a0-20bd29c22df0"
  },
  "convert": {
    "lun": {
      "name": "/vol/volume1/lun1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
  },
  "create_time": "2018-06-04 15:00:00 -0400",
  "location": {
    "namespace": "namespace1",
    "node": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "node1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "qtree": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      }
    }
  }
}
```

```

    }
  },
  "id": 1,
  "name": "qt1"
},
"volume": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "volume1",
  "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
}
},
"metric": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "duration": "PT15S",
  "iops": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "latency": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "status": "ok",
  "throughput": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "timestamp": "2017-01-25 06:20:13 -0500"
},
"name": "/vol/volume1/mtree1/namespace1",
"os_type": "string",
"provisioning_options": {
  "qos_policy": {
    "_links": {
      "self": {

```

```

        "href": "/api/resourcelink"
    },
    },
    "name": "performance",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"snapshot_policy": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    },
    "name": "default",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"storage_service": {
    "name": "string"
},
"tiering": {
    "control": "string",
    "object_stores": [
        {
            "name": "string"
        }
    ],
    "policy": "string"
}
},
"space": {
    "block_size": 512,
    "efficiency_ratio": 2.5,
    "physical_used": 1073741824,
    "physical_used_by_snapshots": 1073741824,
    "size": 1073741824,
    "used": 0
},
"statistics": {
    "iops_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "latency_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    }
}

```

```

    },
    "status": "ok",
    "throughput_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
},
"status": {
    "container_state": "string",
    "state": "online"
},
"subsystem_map": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "anagrpId": "00103050h",
    "nsid": "00000001h",
    "subsystem": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "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"
                }
            }
        ]
    }
}

```

```

    }
  ],
  "name": "subsystem1",
  "os_type": "string",
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
},
"svm": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "svm1",
  "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
},
"uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}

```

## Error

Status: Default

## ONTAP Error Response Codes

Error Code	Description
72090006	The specified namespace was not found.
72090007	The specified namespace was not found.

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

Name	Type	Description
error	<a href="#">returned_error</a>	



### 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	<a href="#">href</a>	

source

The source NVMe namespace for a namespace clone operation. This can be specified using property `clone.source.uuid` or `clone.source.name`. If both properties are supplied, they must refer to the same namespace.

Valid in POST to create a new NVMe namespace as a clone of the source.

Valid in PATCH to overwrite an existing NVMe namespace's data as a clone of another.

Name	Type	Description
name	string	<p>The name of the clone source NVMe namespace. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.</p> <p>NVMe namespace names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;namespace&gt;"</code> where the qtree name is optional.</p> <p>Valid in POST and PATCH.</p>
uuid	string	<p>The unique identifier of the clone source NVMe namespace. Valid in POST and PATCH.</p>

clone

This sub-object is used in POST to create a new NVMe namespace as a clone of an existing namespace, or PATCH to overwrite an existing namespace as a clone of another. Setting a property in this sub-object indicates that a namespace clone is desired.

When used in a PATCH, the patched NVMe namespace's data is over-written as a clone of the source and the following properties are preserved from the patched namespace unless otherwise specified as

part of the PATCH: `auto_delete` (unless specified in the request), `subsystem_map`, `status.state`, and `uuid`.

Name	Type	Description
source	<a href="#">source</a>	<p>The source NVMe namespace for a namespace clone operation. This can be specified using property <code>clone.source.uuid</code> or <code>clone.source.name</code>. If both properties are supplied, they must refer to the same namespace.</p> <p>Valid in POST to create a new NVMe namespace as a clone of the source.</p> <p>Valid in PATCH to overwrite an existing NVMe namespace's data as a clone of another.</p>

#### consistency\_group

The namespace's consistency group. This property is populated for namespaces that are members of a consistency group. If the namespace is a member of a child consistency group, the parent consistency group is reported. A namespace's consistency group is the consistency group of its containing volume.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
name	string	The name of the consistency group.
uuid	string	The unique identifier of the consistency group.

#### lun

The source LUN for convert operation. This can be specified using property `convert.lun.uuid` or `convert.lun.name`. If both properties are supplied, they must refer to the same LUN.

Valid in POST. A convert request from LUN to NVMe namespace cannot be combined with setting any other namespace properties. All other properties of the converted NVMe namespace come from the source LUN.

Name	Type	Description
name	string	The name of the source LUN. Valid in POST. A LUN is located within a volume. Optionally, it can be located within a qtree in a volume.  LUN names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;lun&gt;"</code> where the qtree name is optional.
uuid	string	The unique identifier of the source LUN. Valid in POST.

## convert

This sub-object is used in POST to convert a valid in-place LUN to an NVMe namespace. Setting a property in this sub-object indicates that a conversion from the specified LUN to NVMe namespace is desired.

Name	Type	Description
lun	<a href="#">lun</a>	The source LUN for convert operation. This can be specified using property <code>convert.lun.uuid</code> or <code>convert.lun.name</code> . If both properties are supplied, they must refer to the same LUN.  Valid in POST. A convert request from LUN to NVMe namespace cannot be combined with setting any other namespace properties. All other properties of the converted NVMe namespace come from the source LUN.

## node

The cluster node that hosts the NVMe namespace.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
name	string	
uuid	string	

## qtree

The qtree in which the NVMe namespace is optionally located. Valid in POST.

If properties `name` and `location.qtree.name` and/or `location.qtree.uuid` are specified in the same request, they must refer to the same qtree.

NVMe namespaces do not support rename.

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
<code>id</code>	integer	The identifier for the qtree, unique within the qtree's volume.
<code>name</code>	string	The name of the qtree.

volume

The volume in which the NVMe namespace is located. Valid in POST.

If properties `name` and `location.volume.name` and/or `location.volume.uuid` are specified in the same request, they must refer to the same volume.

NVMe namespaces do not support movement between volumes.

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
<code>name</code>	string	The name of the volume. This field cannot be specified in a PATCH method.
<code>uuid</code>	string	Unique identifier for the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move. <ul style="list-style-type: none"><li>• example: 028baa66-41bd-11e9-81d5-00a0986138f7</li><li>• Introduced in: 9.6</li><li>• x-nullable: true</li></ul>

location

The location of the NVMe namespace within the ONTAP cluster. NVMe namespaces do not support rename, or movement between volumes. Valid in POST.

Name	Type	Description
namespace	string	<p>The base name component of the NVMe namespace. Valid in POST.</p> <p>If properties <code>name</code> and <code>location.namespace</code> are specified in the same request, they must refer to the base name.</p> <p>NVMe namespaces do not support rename.</p>
node	<a href="#">node</a>	The cluster node that hosts the NVMe namespace.
qtree	<a href="#">qtree</a>	<p>The qtree in which the NVMe namespace is optionally located. Valid in POST.</p> <p>If properties <code>name</code> and <code>location.qtree.name</code> and/or <code>location.qtree.uuid</code> are specified in the same request, they must refer to the same qtree.</p> <p>NVMe namespaces do not support rename.</p>
volume	<a href="#">volume</a>	<p>The volume in which the NVMe namespace is located. Valid in POST.</p> <p>If properties <code>name</code> and <code>location.volume.name</code> and/or <code>location.volume.uuid</code> are specified in the same request, they must refer to the same volume.</p> <p>NVMe namespaces do not support movement between volumes.</p>

iops

The rate of I/O operations observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### latency

The round trip latency in microseconds observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### throughput

The rate of throughput bytes per second observed at the storage object.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.

Name	Type	Description
write	integer	Performance metric for write I/O operations.

metric

Performance numbers, such as IOPS latency and throughput

Name	Type	Description
_links	<a href="#">_links</a>	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
iops	<a href="#">iops</a>	The rate of I/O operations observed at the storage object.
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.



Name	Type	Description
throughput	<a href="#">throughput</a>	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

`self_link`

Name	Type	Description
self	<a href="#">href</a>	

`qos_policy`

The QoS policy for the volume provisioned to host the namespace. This property is only supported when the request provisions a new volume. If no `qos_policy` is provided, a policy is be set based on the `storage_service.name`, which defaults to the most performant service available.

Name	Type	Description
<code>_links</code>	<a href="#">self_link</a>	
name	string	The QoS policy group name. This is mutually exclusive with UUID and other QoS attributes during POST and PATCH.
uuid	string	The QoS policy group UUID. This is mutually exclusive with name and other QoS attributes during POST and PATCH.

`snapshot_policy_reference`

This is a reference to the snapshot policy.

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
name	string	
uuid	string	

`storage_service`

Determines the placement of the namespace based on the value specified. Valid in POST.

Name	Type	Description
name	string	Storage service name. If not specified, the default value is the most performant for the platform.

#### object\_stores

Name	Type	Description
name	string	The name of the object store to use. Used for placement.

#### tiering

The tiering placement and policy definitions for the volume provisioned to host the namespace. This property is only supported when the request provisions a new volume.

Name	Type	Description
control	string	Storage tiering placement rules for the object.
object_stores	array[ <a href="#">object_stores</a> ]	Object stores to use. Used for placement.

Name	Type	Description
policy	string	<p>Policy that determines whether the user data blocks of a volume in a FabricPool will be tiered to the cloud store when they become cold.</p> <p>FabricPool combines flash (performance tier) with a cloud store into a single aggregate. Temperature of a volume block increases if it is accessed frequently and decreases when it is not. Valid in POST or PATCH.</p> <p>all &amp;dash; Allows tiering of both snapshots and active file system user data to the cloud store as soon as possible by ignoring the temperature on the volume blocks.</p> <p>auto &amp;dash; Allows tiering of both snapshot and active file system user data to the cloud store</p> <p>none &amp;dash; Volume blocks are not be tiered to the cloud store.</p> <p>snapshot_only &amp;dash; Allows tiering of only the volume snapshots not associated with the active file system.</p> <p>The default tiering policy is "snapshot-only" for a FlexVol volume and "none" for a FlexGroup volume. The default minimum cooling period for the "snapshot-only" tiering policy is 2 days and for the "auto" tiering policy it is 31 days.</p>

provisioning\_options

Options that are applied to the operation.



Name	Type	Description
auto	boolean	<p>If the volume specified in the request does not exist, automatically provision one of appropriate size. If the volume does exist, resize it to accommodate the new namespace.</p> <p>This property is only supported on Unified ONTAP.</p> <p>The following behavior is different from a traditional POST request:</p> <ul style="list-style-type: none"> <li>• The operation is asynchronous.</li> <li>• The <code>qos_policy</code> property is supported and is applied to the provisioned volume. A default QoS policy is applied to the volume if one is not provided.</li> <li>• The <code>provisioning_options.count</code> property is supported, provisioning <i>count</i> namespaces on the volume using the specified properties.</li> <li>• The <code>subsystem_map</code> property is supported. If the specified subsystem does not exist, it is created. The namespace is mapped to this subsystem. If a subsystem is provisioned in this way, it is deleted after it is no longer mapped to any namespaces.</li> <li>• The <code>clone</code> and <code>convert</code> properties are not supported.</li> <li>• When performing <code>records</code> based operations, specifying this property in the query applies to the entire operation. Specifying it for an individual record within the request applies to only that record.</li> <li>• Many other <code>provisioning_options</code> properties are supported to control the placement of the namespace and the properties of the volume</li> </ul>

Name	Type	Description
count	integer	The number of namespaces to provision with these properties. Only POST requests based on <code>space.size</code> are supported. When provided, the name is considered a prefix, and a suffix of the form <code>_&lt;N&gt;</code> is generated where N is the next available numeric index, starting with 1.
qos_policy	<a href="#">qos_policy</a>	The QoS policy for the volume provisioned to host the namespace. This property is only supported when the request provisions a new volume. If no <code>qos_policy</code> is provided, a policy is be set based on the <code>storage_service.name</code> , which defaults to the most performant service available.
snapshot_policy	<a href="#">snapshot_policy_reference</a>	This is a reference to the snapshot policy.
storage_service	<a href="#">storage_service</a>	Determines the placement of the namespace based on the value specified. Valid in POST.
tiering	<a href="#">tiering</a>	The tiering placement and policy definitions for the volume provisioned to host the namespace. This property is only supported when the request provisions a new volume.
use_mirrored_aggregates	boolean	Specifies whether mirrored aggregates are selected when provisioning the volume to host the namespace. Only mirrored aggregates are used if this parameter is set to <i>true</i> and only unmirrored aggregates are used if this parameter is set to <i>false</i> . The default value is <i>true</i> for a MetroCluster configuration and is <i>false</i> for a non-MetroCluster configuration.

guarantee

Properties that request and report the space guarantee for the NVMe namespace.

Name	Type	Description
requested	boolean	<p>The requested space reservation policy for the NVMe namespace. If <i>true</i>, a space reservation is requested for the namespace; if <i>false</i>, the namespace is thin provisioned. Guaranteeing a space reservation request for a namespace requires that the volume in which the namespace resides also be space reserved and that the fractional reserve for the volume be 100%.</p> <p>The space reservation policy for an NVMe namespace is determined by ONTAP.</p> <ul style="list-style-type: none"><li>• readOnly: 1</li><li>• Introduced in: 9.6</li><li>• x-nullable: true</li></ul>
reserved	boolean	<p>Reports if the NVMe namespace is space guaranteed.</p> <p>This property is <i>true</i> if a space guarantee is requested and the containing volume and aggregate support the request. This property is <i>false</i> if a space guarantee is not requested or if a space guarantee is requested and either the containing volume and aggregate do not support the request.</p>

## space

The storage space related properties of the NVMe namespace.

Name	Type	Description
block_size	integer	<p>The size of blocks in the namespace in bytes. The default for namespaces with an <code>os_type</code> of <i>vmware</i> is 512. All other namespaces default to 4096.</p> <p>Valid in POST when creating an NVMe namespace that is not a clone of another. Disallowed in POST when creating a namespace clone. Valid in POST.</p>
efficiency_ratio	number	<p>The storage efficiency ratio of the namespace without snapshots. (Logical Used / Used) This property is not available on the namespace object in the REST API and is not reported for GET requests. See the containing volume object for this information.</p>
guarantee	<a href="#">guarantee</a>	<p>Properties that request and report the space guarantee for the NVMe namespace.</p>
physical_used	integer	<p>The number of bytes consumed on the disk by the namespace, excluding snapshots. This property is not available on the namespace object in the REST API and is not reported for GET requests. See the containing volume object for this information.</p>
physical_used_by_snapshots	integer	<p>The number of bytes consumed on the disk by the namespace's snapshots. This property is not available on the namespace object in the REST API and is not reported for GET requests. See the containing volume object for this information.</p>



Name	Type	Description
size	integer	<p>The total provisioned size of the NVMe namespace. Valid in POST and PATCH. The NVMe namespace size can be increased but not be made smaller using the REST interface.</p> <p>The maximum and minimum sizes listed here are the absolute maximum and absolute minimum sizes in bytes. The maximum size is variable with respect to large NVMe namespace support in ONTAP. If large namespaces are supported, the maximum size is 128 TB (140737488355328 bytes) and if not supported, the maximum size is just under 16 TB (17557557870592 bytes). The minimum size supported is always 4096 bytes.</p> <p>For more information, see <i>Size properties</i> in the <i>docs</i> section of the ONTAP REST API documentation.</p> <ul style="list-style-type: none"> <li>• example: 1073741824</li> <li>• format: int64</li> <li>• Max value: 140737488355328</li> <li>• Min value: 4096</li> <li>• Introduced in: 9.6</li> <li>• x-nullable: true</li> </ul>

Name	Type	Description
used	integer	<p>The amount of space consumed by the main data stream of the NVMe namespace.</p> <p>This value is the total space consumed in the volume by the NVMe namespace, including filesystem overhead, but excluding prefix and suffix streams. Due to internal filesystem overhead and the many ways NVMe filesystems and applications utilize blocks within a namespace, this value does not necessarily reflect actual consumption/availability from the perspective of the filesystem or application. Without specific knowledge of how the namespace blocks are utilized outside of ONTAP, this property should not be used and an indicator for an out-of-space condition.</p> <p>For more information, see <i>Size properties</i> in the <i>docs</i> section of the ONTAP REST API documentation.</p> <ul style="list-style-type: none"> <li>• format: int64</li> <li>• readOnly: 1</li> <li>• Introduced in: 9.6</li> <li>• x-nullable: true</li> </ul>

#### iops\_raw

The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.

Name	Type	Description
other	integer	<p>Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.</p>

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### latency\_raw

The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### throughput\_raw

Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## statistics

These are raw performance numbers, such as IOPS latency and throughput. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.

Name	Type	Description
iops_raw	<a href="#">iops_raw</a>	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.
latency_raw	<a href="#">latency_raw</a>	The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput_raw	<a href="#">throughput_raw</a>	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.

Name	Type	Description
timestamp	string	The timestamp of the performance data.

status

Status information about the NVMe namespace.

Name	Type	Description
container_state	string	The state of the volume and aggregate that contain the NVMe namespace. Namespaces are only available when their containers are available.
mapped	boolean	Reports if the NVMe namespace is mapped to an NVMe subsystem.  There is an added computational cost to retrieving this property's value. It is not populated for a GET request unless it is explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more.
read_only	boolean	Reports if the NVMe namespace allows only read access.
state	string	The state of the NVMe namespace. Normal states for a namespace are <i>online</i> and <i>offline</i> . Other states indicate errors.

consistency\_group\_nvme\_host\_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	<p>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.</p> <p>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.</p>
group_size	string	<p>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.</p>
hash_function	string	<p>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.</p>

Name	Type	Description
host_secret_key	string	<p>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.</p> <p>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.</p>
mode	string	<p>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:</p> <ul style="list-style-type: none"> <li>• none: The host has neither the host nor controller secret configured, and no authentication is performed.</li> <li>• unidirectional: The host has a host secret configured. The controller will authenticate the host.</li> <li>• bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.</li> </ul>

tls

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

Name	Type	Description
configured_psk	string	<p>A user supplied pre-shared key (PSK) value in PSK Interchange Format. Optional in POST.</p> <p>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>.</p> <p>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.</p>



Name	Type	Description
key_type	string	<p>The method by which the TLS pre-shared key (PSK) is configured for the host. Optional in POST.</p> <p>The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• <code>none</code> - TLS is not configured for the host connection. No value is allowed for property <code>configured_psk</code>.</li> <li>• <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.</li> </ul> <p>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>.</p>

## hosts

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

Name	Type	Description
dh_hmac_chap	<a href="#">consistency_group_nvme_host_dh_hmac_chap</a>	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 io_queue count and I/O queue depth are being used.
tls	<a href="#">tls</a>	A container for the configuration for NVMe/TCP-TLS transport session for the host.

### subsystem

The NVMe subsystem to which the NVMe namespace is mapped.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
comment	string	A configurable comment for the NVMe subsystem. Optional in POST.
hosts	array[ <a href="#">hosts</a> ]	The NVMe hosts configured for access to the NVMe subsystem. Optional in POST.
name	string	The name of the NVMe subsystem. Once created, an NVMe subsystem cannot be renamed. Optional in POST.
os_type	string	The host operating system of the NVMe subsystem's hosts. Optional in POST.
uuid	string	The unique identifier of the NVMe subsystem.

### subsystem\_map

The NVMe subsystem with which the NVMe namespace is associated. A namespace can be mapped to zero (0) or one (1) subsystems.

There is an added computational cost to retrieving property values for `subsystem_map`. They are not populated for a GET request unless explicitly requested using the `fields` query parameter. See [Requesting specific fields](#) to learn more. These properties are supported only for GET.

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

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
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
message	string	Message argument

returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

## Update an NVMe namespace

PATCH `/storage/namespaces/{uuid}`

**Introduced In:** 9.6

Updates an NVMe namespace.

### Related ONTAP commands

- `volume file clone autodelete`
- `vserver nvme namespace modify`

### Learn more

- [DOC /storage/namespaces](#)

### Parameters

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

Name	Type	In	Required	Description
return_timeout	integer	query	False	<p>The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> <li>• Max value: 120</li> <li>• Min value: 0</li> </ul>

### Request Body

Name	Type	Description
_links	<a href="#">_links</a>	

Name	Type	Description
auto_delete	boolean	<p>This property marks the NVMe namespace for auto deletion when the volume containing the namespace runs out of space. This is most commonly set on namespace clones.</p> <p>When set to <i>true</i>, the NVMe namespace becomes eligible for automatic deletion when the volume runs out of space. Auto deletion only occurs when the volume containing the namespace is also configured for auto deletion and free space in the volume decreases below a particular threshold.</p> <p>This property is optional in POST and PATCH. The default value for a new NVMe namespace is <i>false</i>.</p> <p>There is an added computational cost to retrieving this property's value. It is not populated for a GET request unless it is explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more.</p>
clone	<a href="#">clone</a>	<p>This sub-object is used in POST to create a new NVMe namespace as a clone of an existing namespace, or PATCH to overwrite an existing namespace as a clone of another. Setting a property in this sub-object indicates that a namespace clone is desired.</p> <p>When used in a PATCH, the patched NVMe namespace's data is over-written as a clone of the source and the following properties are preserved from the patched namespace unless otherwise specified as part of the PATCH: <code>auto_delete</code> (unless specified in the request), <code>subsystem_map</code>, <code>status.state</code>, and <code>uuid</code>.</p>

Name	Type	Description
comment	string	A configurable comment available for use by the administrator. Valid in POST and PATCH.
consistency_group	<a href="#">consistency_group</a>	The namespace's consistency group. This property is populated for namespaces that are members of a consistency group. If the namespace is a member of a child consistency group, the parent consistency group is reported. A namespace's consistency group is the consistency group of its containing volume.
convert	<a href="#">convert</a>	This sub-object is used in POST to convert a valid in-place LUN to an NVMe namespace. Setting a property in this sub-object indicates that a conversion from the specified LUN to NVMe namespace is desired.
create_time	string	The time the NVMe namespace was created.
enabled	boolean	The enabled state of the NVMe namespace. Certain error conditions cause the namespace to become disabled. If the namespace is disabled, check the <code>status.state</code> property to determine what error disabled the namespace. An NVMe namespace is enabled automatically when it is created.
location	<a href="#">location</a>	<p>The location of the NVMe namespace within the ONTAP cluster. NVMe namespaces do not support rename, or movement between volumes. Valid in POST.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.6</li> <li>• readCreate: 1</li> </ul>
metric	<a href="#">metric</a>	Performance numbers, such as IOPS latency and throughput

Name	Type	Description
name	string	<p>The name of the NVMe namespace. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.</p> <p>NVMe namespace names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;namespace&gt;"</code> where the qtree name is optional.</p> <p>Renaming an NVMe namespace is not supported. Valid in POST.</p>
os_type	string	<p>The operating system type of the NVMe namespace.</p> <p>Required in POST when creating an NVMe namespace that is not a clone of another. Disallowed in POST when creating a namespace clone.</p>
provisioning_options	<a href="#">provisioning_options</a>	Options that are applied to the operation.
space	<a href="#">space</a>	The storage space related properties of the NVMe namespace.
statistics	<a href="#">statistics</a>	These are raw performance numbers, such as IOPS latency and throughput. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.
status	<a href="#">status</a>	Status information about the NVMe namespace.



Name	Type	Description
subsystem_map	<a href="#">subsystem_map</a>	<p>The NVMe subsystem with which the NVMe namespace is associated. A namespace can be mapped to zero (0) or one (1) subsystems.</p> <p>There is an added computational cost to retrieving property values for <code>subsystem_map</code>. They are not populated for a GET request unless explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more. These properties are supported only for GET.</p>
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.
uuid	string	The unique identifier of the NVMe namespace.

## Example request

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "clone": {
    "source": {
      "name": "/vol/volume1/namespace1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
  },
  "comment": "string",
  "consistency_group": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "cgl",
    "uuid": "4abc2317-4332-9d37-93a0-20bd29c22df0"
  },
  "convert": {
    "lun": {
      "name": "/vol/volume1/lun1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
  },
  "create_time": "2018-06-04 15:00:00 -0400",
  "location": {
    "namespace": "namespace1",
    "node": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "node1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "qtree": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      }
    }
  }
}
```

```

    }
  },
  "id": 1,
  "name": "qt1"
},
"volume": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "volume1",
  "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
}
},
"metric": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "duration": "PT15S",
  "iops": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "latency": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "status": "ok",
  "throughput": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "timestamp": "2017-01-25 06:20:13 -0500"
},
"name": "/vol/volume1/mtree1/namespace1",
"os_type": "string",
"provisioning_options": {
  "qos_policy": {
    "_links": {
      "self": {

```

```

        "href": "/api/resourcelink"
    },
    },
    "name": "performance",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"snapshot_policy": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    },
    "name": "default",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"storage_service": {
    "name": "string"
},
"tiering": {
    "control": "string",
    "object_stores": [
        {
            "name": "string"
        }
    ],
    "policy": "string"
}
},
"space": {
    "block_size": 512,
    "efficiency_ratio": 2.5,
    "physical_used": 1073741824,
    "physical_used_by_snapshots": 1073741824,
    "size": 1073741824,
    "used": 0
},
"statistics": {
    "iops_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "latency_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    }
}

```

```

    },
    "status": "ok",
    "throughput_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
},
"status": {
    "container_state": "string",
    "state": "online"
},
"subsystem_map": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "anagrpId": "00103050h",
    "nsid": "00000001h",
    "subsystem": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "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"
                }
            }
        ]
    }
}

```

```

    }
  ],
  "name": "subsystem1",
  "os_type": "string",
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
},
"svm": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "svm1",
  "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
},
"uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}

```

## Response

Status: 200, Ok

## Response

Status: 202, Accepted

## Error

Status: Default

## ONTAP Error Response Codes

Error Code	Description
5374127	The specified namespace name is invalid.
5376461	The specified namespace name is invalid.
5376462	The specified namespace name is too long.
5376463	The snapshot portion of the specified namespace name is too long.

Error Code	Description
5376466	An attempt was made to rename an NVMe namespace to a snapshot name.
5376467	An attempt was made to rename a primary NVMe namespace to a secondary name.
5376468	An attempt was made to rename an NVMe namespace to a reserved name.
13565952	The namespace clone request failed.
72089724	The specified namespace size is too large.
72089730	The specified namespace cannot be updated as it resides in a snapshot.
72090005	The specified <code>clone.source.uuid</code> and <code>clone.source.name</code> do not refer to the same LUN.
72090006	The specified namespace was not found. This can apply to <code>clone.source</code> or the target namespace. The <code>target</code> property of the error object identifies the property.
72090007	The specified namespace was not found. This can apply to <code>clone.source</code> or the target namespace. The <code>target</code> property of the error object identifies the property.
72090010	An error occurred after successfully overwriting data for the namespace as a clone. Some properties were not modified.
72090011	An error occurred after successfully modifying some of the properties of the namespace. Some properties were not modified.
72090016	The namespace's aggregate is offline. The aggregate must be online to modify or remove the namespace.
72090017	The namespace's volume is offline. The volume must be online to modify or remove the namespace.
72090038	An attempt was made to reduce the size of the specified namespace.

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

Name	Type	Description
error	<a href="#">returned_error</a>	

### 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	<a href="#">href</a>	

source

The source NVMe namespace for a namespace clone operation. This can be specified using property `clone.source.uuid` or `clone.source.name`. If both properties are supplied, they must refer to the same namespace.

Valid in POST to create a new NVMe namespace as a clone of the source.

Valid in PATCH to overwrite an existing NVMe namespace's data as a clone of another.

Name	Type	Description
name	string	<p>The name of the clone source NVMe namespace. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.</p> <p>NVMe namespace names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;namespace&gt;"</code> where the qtree name is optional.</p> <p>Valid in POST and PATCH.</p>
uuid	string	<p>The unique identifier of the clone source NVMe namespace. Valid in POST and PATCH.</p>

clone

This sub-object is used in POST to create a new NVMe namespace as a clone of an existing namespace, or PATCH to overwrite an existing namespace as a clone of another. Setting a property in this sub-object indicates that a namespace clone is desired.

When used in a PATCH, the patched NVMe namespace's data is over-written as a clone of the source and the following properties are preserved from the patched namespace unless otherwise specified as

part of the PATCH: `auto_delete` (unless specified in the request), `subsystem_map`, `status.state`, and `uuid`.

Name	Type	Description
source	<a href="#">source</a>	<p>The source NVMe namespace for a namespace clone operation. This can be specified using property <code>clone.source.uuid</code> or <code>clone.source.name</code>. If both properties are supplied, they must refer to the same namespace.</p> <p>Valid in POST to create a new NVMe namespace as a clone of the source.</p> <p>Valid in PATCH to overwrite an existing NVMe namespace's data as a clone of another.</p>

#### consistency\_group

The namespace's consistency group. This property is populated for namespaces that are members of a consistency group. If the namespace is a member of a child consistency group, the parent consistency group is reported. A namespace's consistency group is the consistency group of its containing volume.

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
name	string	The name of the consistency group.
uuid	string	The unique identifier of the consistency group.

#### lun

The source LUN for convert operation. This can be specified using property `convert.lun.uuid` or `convert.lun.name`. If both properties are supplied, they must refer to the same LUN.

Valid in POST. A convert request from LUN to NVMe namespace cannot be combined with setting any other namespace properties. All other properties of the converted NVMe namespace come from the source LUN.

Name	Type	Description
name	string	The name of the source LUN. Valid in POST. A LUN is located within a volume. Optionally, it can be located within a qtree in a volume.  LUN names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;lun&gt;"</code> where the qtree name is optional.
uuid	string	The unique identifier of the source LUN. Valid in POST.

## convert

This sub-object is used in POST to convert a valid in-place LUN to an NVMe namespace. Setting a property in this sub-object indicates that a conversion from the specified LUN to NVMe namespace is desired.

Name	Type	Description
lun	<a href="#">lun</a>	The source LUN for convert operation. This can be specified using property <code>convert.lun.uuid</code> or <code>convert.lun.name</code> . If both properties are supplied, they must refer to the same LUN.  Valid in POST. A convert request from LUN to NVMe namespace cannot be combined with setting any other namespace properties. All other properties of the converted NVMe namespace come from the source LUN.

## node

The cluster node that hosts the NVMe namespace.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
name	string	
uuid	string	

## qtree

The qtree in which the NVMe namespace is optionally located. Valid in POST.

If properties `name` and `location.qtree.name` and/or `location.qtree.uuid` are specified in the same request, they must refer to the same qtree.

NVMe namespaces do not support rename.

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
<code>id</code>	integer	The identifier for the qtree, unique within the qtree's volume.
<code>name</code>	string	The name of the qtree.

volume

The volume in which the NVMe namespace is located. Valid in POST.

If properties `name` and `location.volume.name` and/or `location.volume.uuid` are specified in the same request, they must refer to the same volume.

NVMe namespaces do not support movement between volumes.

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
<code>name</code>	string	The name of the volume. This field cannot be specified in a PATCH method.
<code>uuid</code>	string	Unique identifier for the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move. <ul style="list-style-type: none"><li>• example: 028baa66-41bd-11e9-81d5-00a0986138f7</li><li>• Introduced in: 9.6</li><li>• x-nullable: true</li></ul>

location

The location of the NVMe namespace within the ONTAP cluster. NVMe namespaces do not support rename, or movement between volumes. Valid in POST.

Name	Type	Description
namespace	string	<p>The base name component of the NVMe namespace. Valid in POST.</p> <p>If properties <code>name</code> and <code>location.namespace</code> are specified in the same request, they must refer to the base name.</p> <p>NVMe namespaces do not support rename.</p>
node	<a href="#">node</a>	The cluster node that hosts the NVMe namespace.
qtree	<a href="#">qtree</a>	<p>The qtree in which the NVMe namespace is optionally located. Valid in POST.</p> <p>If properties <code>name</code> and <code>location.qtree.name</code> and/or <code>location.qtree.uuid</code> are specified in the same request, they must refer to the same qtree.</p> <p>NVMe namespaces do not support rename.</p>
volume	<a href="#">volume</a>	<p>The volume in which the NVMe namespace is located. Valid in POST.</p> <p>If properties <code>name</code> and <code>location.volume.name</code> and/or <code>location.volume.uuid</code> are specified in the same request, they must refer to the same volume.</p> <p>NVMe namespaces do not support movement between volumes.</p>

iops

The rate of I/O operations observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### latency

The round trip latency in microseconds observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### throughput

The rate of throughput bytes per second observed at the storage object.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.

Name	Type	Description
write	integer	Performance metric for write I/O operations.

metric

Performance numbers, such as IOPS latency and throughput

Name	Type	Description
_links	<a href="#">_links</a>	
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
iops	<a href="#">iops</a>	The rate of I/O operations observed at the storage object.
latency	<a href="#">latency</a>	The round trip latency in microseconds observed at the storage object.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.

Name	Type	Description
throughput	<a href="#">throughput</a>	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

self\_link

Name	Type	Description
self	<a href="#">href</a>	

qos\_policy

The QoS policy for the volume provisioned to host the namespace. This property is only supported when the request provisions a new volume. If no `qos_policy` is provided, a policy is be set based on the `storage_service.name`, which defaults to the most performant service available.

Name	Type	Description
_links	<a href="#">self_link</a>	
name	string	The QoS policy group name. This is mutually exclusive with UUID and other QoS attributes during POST and PATCH.
uuid	string	The QoS policy group UUID. This is mutually exclusive with name and other QoS attributes during POST and PATCH.

snapshot\_policy\_reference

This is a reference to the snapshot policy.

Name	Type	Description
_links	<a href="#">_links</a>	
name	string	
uuid	string	

storage\_service

Determines the placement of the namespace based on the value specified. Valid in POST.



Name	Type	Description
name	string	Storage service name. If not specified, the default value is the most performant for the platform.

#### object\_stores

Name	Type	Description
name	string	The name of the object store to use. Used for placement.

#### tiering

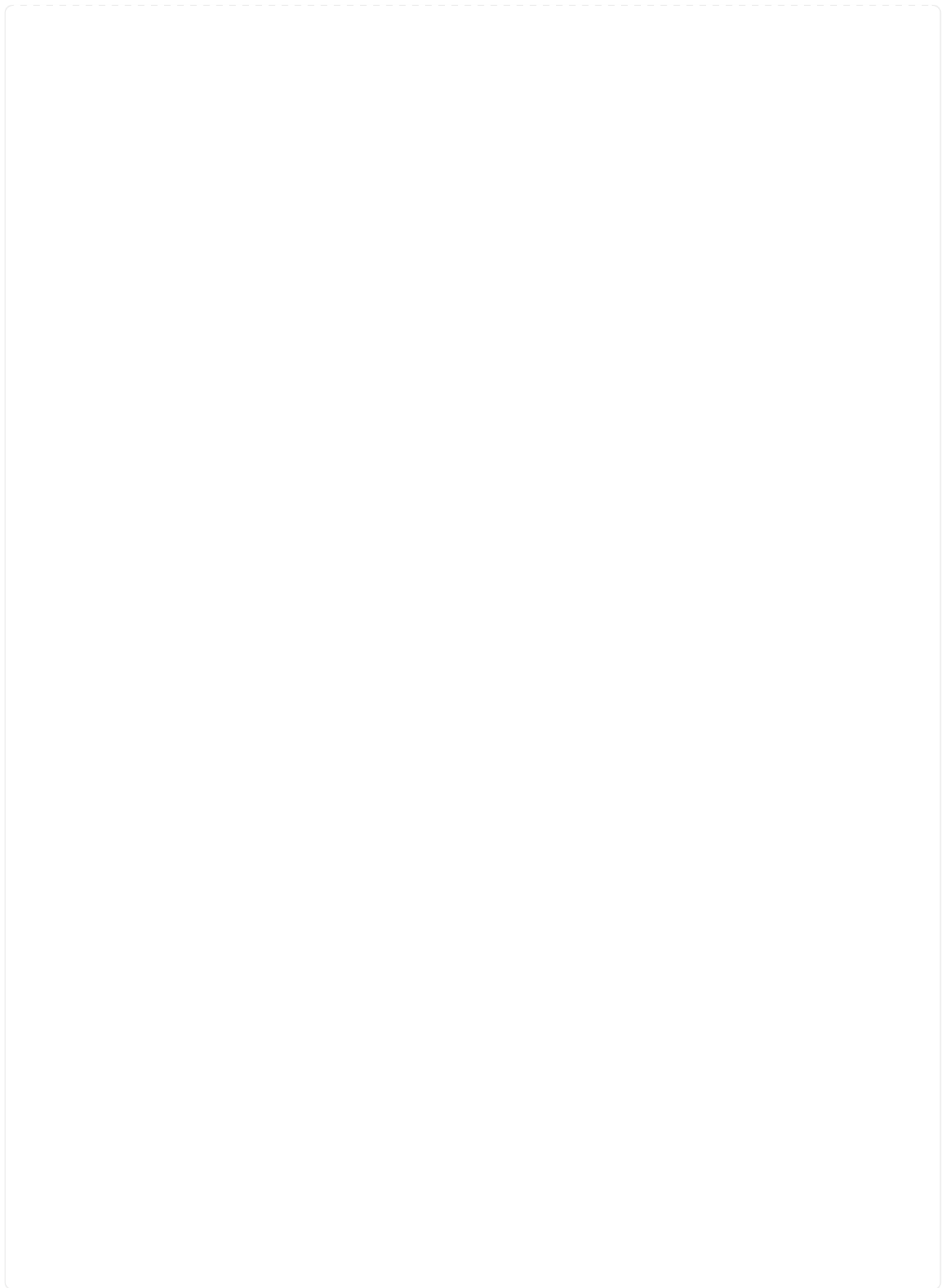
The tiering placement and policy definitions for the volume provisioned to host the namespace. This property is only supported when the request provisions a new volume.

Name	Type	Description
control	string	Storage tiering placement rules for the object.
object_stores	array[ <a href="#">object_stores</a> ]	Object stores to use. Used for placement.

Name	Type	Description
policy	string	<p>Policy that determines whether the user data blocks of a volume in a FabricPool will be tiered to the cloud store when they become cold.</p> <p>FabricPool combines flash (performance tier) with a cloud store into a single aggregate. Temperature of a volume block increases if it is accessed frequently and decreases when it is not. Valid in POST or PATCH.</p> <p>all &amp;dash; Allows tiering of both snapshots and active file system user data to the cloud store as soon as possible by ignoring the temperature on the volume blocks.</p> <p>auto &amp;dash; Allows tiering of both snapshot and active file system user data to the cloud store</p> <p>none &amp;dash; Volume blocks are not be tiered to the cloud store.</p> <p>snapshot_only &amp;dash; Allows tiering of only the volume snapshots not associated with the active file system.</p> <p>The default tiering policy is "snapshot-only" for a FlexVol volume and "none" for a FlexGroup volume. The default minimum cooling period for the "snapshot-only" tiering policy is 2 days and for the "auto" tiering policy it is 31 days.</p>

provisioning\_options

Options that are applied to the operation.



Name	Type	Description
auto	boolean	<p>If the volume specified in the request does not exist, automatically provision one of appropriate size. If the volume does exist, resize it to accommodate the new namespace.</p> <p>This property is only supported on Unified ONTAP.</p> <p>The following behavior is different from a traditional POST request:</p> <ul style="list-style-type: none"> <li>• The operation is asynchronous.</li> <li>• The <code>qos_policy</code> property is supported and is applied to the provisioned volume. A default QoS policy is applied to the volume if one is not provided.</li> <li>• The <code>provisioning_options.count</code> property is supported, provisioning <i>count</i> namespaces on the volume using the specified properties.</li> <li>• The <code>subsystem_map</code> property is supported. If the specified subsystem does not exist, it is created. The namespace is mapped to this subsystem. If a subsystem is provisioned in this way, it is deleted after it is no longer mapped to any namespaces.</li> <li>• The <code>clone</code> and <code>convert</code> properties are not supported.</li> <li>• When performing <code>records</code> based operations, specifying this property in the query applies to the entire operation. Specifying it for an individual record within the request applies to only that record.</li> <li>• Many other <code>provisioning_options</code> properties are supported to control the placement of the namespace and the properties of the volume</li> </ul>

Name	Type	Description
count	integer	The number of namespaces to provision with these properties. Only POST requests based on <code>space.size</code> are supported. When provided, the name is considered a prefix, and a suffix of the form <code>_&lt;N&gt;</code> is generated where N is the next available numeric index, starting with 1.
qos_policy	<a href="#">qos_policy</a>	The QoS policy for the volume provisioned to host the namespace. This property is only supported when the request provisions a new volume. If no <code>qos_policy</code> is provided, a policy is be set based on the <code>storage_service.name</code> , which defaults to the most performant service available.
snapshot_policy	<a href="#">snapshot_policy_reference</a>	This is a reference to the snapshot policy.
storage_service	<a href="#">storage_service</a>	Determines the placement of the namespace based on the value specified. Valid in POST.
tiering	<a href="#">tiering</a>	The tiering placement and policy definitions for the volume provisioned to host the namespace. This property is only supported when the request provisions a new volume.
use_mirrored_aggregates	boolean	Specifies whether mirrored aggregates are selected when provisioning the volume to host the namespace. Only mirrored aggregates are used if this parameter is set to <i>true</i> and only unmirrored aggregates are used if this parameter is set to <i>false</i> . The default value is <i>true</i> for a MetroCluster configuration and is <i>false</i> for a non-MetroCluster configuration.

guarantee

Properties that request and report the space guarantee for the NVMe namespace.

Name	Type	Description
requested	boolean	<p>The requested space reservation policy for the NVMe namespace. If <i>true</i>, a space reservation is requested for the namespace; if <i>false</i>, the namespace is thin provisioned. Guaranteeing a space reservation request for a namespace requires that the volume in which the namespace resides also be space reserved and that the fractional reserve for the volume be 100%.</p> <p>The space reservation policy for an NVMe namespace is determined by ONTAP.</p> <ul style="list-style-type: none"><li>• readOnly: 1</li><li>• Introduced in: 9.6</li><li>• x-nullable: true</li></ul>
reserved	boolean	<p>Reports if the NVMe namespace is space guaranteed.</p> <p>This property is <i>true</i> if a space guarantee is requested and the containing volume and aggregate support the request. This property is <i>false</i> if a space guarantee is not requested or if a space guarantee is requested and either the containing volume and aggregate do not support the request.</p>

space

The storage space related properties of the NVMe namespace.

Name	Type	Description
block_size	integer	<p>The size of blocks in the namespace in bytes. The default for namespaces with an <code>os_type</code> of <i>vmware</i> is 512. All other namespaces default to 4096.</p> <p>Valid in POST when creating an NVMe namespace that is not a clone of another. Disallowed in POST when creating a namespace clone. Valid in POST.</p>
efficiency_ratio	number	<p>The storage efficiency ratio of the namespace without snapshots. (Logical Used / Used) This property is not available on the namespace object in the REST API and is not reported for GET requests. See the containing volume object for this information.</p>
guarantee	<a href="#">guarantee</a>	<p>Properties that request and report the space guarantee for the NVMe namespace.</p>
physical_used	integer	<p>The number of bytes consumed on the disk by the namespace, excluding snapshots. This property is not available on the namespace object in the REST API and is not reported for GET requests. See the containing volume object for this information.</p>
physical_used_by_snapshots	integer	<p>The number of bytes consumed on the disk by the namespace's snapshots. This property is not available on the namespace object in the REST API and is not reported for GET requests. See the containing volume object for this information.</p>

Name	Type	Description
size	integer	<p>The total provisioned size of the NVMe namespace. Valid in POST and PATCH. The NVMe namespace size can be increased but not be made smaller using the REST interface.</p> <p>The maximum and minimum sizes listed here are the absolute maximum and absolute minimum sizes in bytes. The maximum size is variable with respect to large NVMe namespace support in ONTAP. If large namespaces are supported, the maximum size is 128 TB (140737488355328 bytes) and if not supported, the maximum size is just under 16 TB (17557557870592 bytes). The minimum size supported is always 4096 bytes.</p> <p>For more information, see <i>Size properties</i> in the <i>docs</i> section of the ONTAP REST API documentation.</p> <ul style="list-style-type: none"> <li>• example: 1073741824</li> <li>• format: int64</li> <li>• Max value: 140737488355328</li> <li>• Min value: 4096</li> <li>• Introduced in: 9.6</li> <li>• x-nullable: true</li> </ul>



Name	Type	Description
used	integer	<p>The amount of space consumed by the main data stream of the NVMe namespace.</p> <p>This value is the total space consumed in the volume by the NVMe namespace, including filesystem overhead, but excluding prefix and suffix streams. Due to internal filesystem overhead and the many ways NVMe filesystems and applications utilize blocks within a namespace, this value does not necessarily reflect actual consumption/availability from the perspective of the filesystem or application. Without specific knowledge of how the namespace blocks are utilized outside of ONTAP, this property should not be used and an indicator for an out-of-space condition.</p> <p>For more information, see <i>Size properties</i> in the <i>docs</i> section of the ONTAP REST API documentation.</p> <ul style="list-style-type: none"> <li>• format: int64</li> <li>• readOnly: 1</li> <li>• Introduced in: 9.6</li> <li>• x-nullable: true</li> </ul>

#### iops\_raw

The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.

Name	Type	Description
other	integer	<p>Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.</p>

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### latency\_raw

The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

#### throughput\_raw

Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

## statistics

These are raw performance numbers, such as IOPS latency and throughput. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.

Name	Type	Description
iops_raw	<a href="#">iops_raw</a>	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.
latency_raw	<a href="#">latency_raw</a>	The raw latency in microseconds observed at the storage object. This should be divided by the raw IOPS value to calculate the average latency per I/O operation.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput_raw	<a href="#">throughput_raw</a>	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.

Name	Type	Description
timestamp	string	The timestamp of the performance data.

status

Status information about the NVMe namespace.

Name	Type	Description
container_state	string	The state of the volume and aggregate that contain the NVMe namespace. Namespaces are only available when their containers are available.
mapped	boolean	Reports if the NVMe namespace is mapped to an NVMe subsystem.  There is an added computational cost to retrieving this property's value. It is not populated for a GET request unless it is explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more.
read_only	boolean	Reports if the NVMe namespace allows only read access.
state	string	The state of the NVMe namespace. Normal states for a namespace are <i>online</i> and <i>offline</i> . Other states indicate errors.

consistency\_group\_nvme\_host\_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	<p>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.</p> <p>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.</p>
group_size	string	<p>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.</p>
hash_function	string	<p>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.</p>

Name	Type	Description
host_secret_key	string	<p>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.</p> <p>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.</p>
mode	string	<p>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:</p> <ul style="list-style-type: none"> <li>• none: The host has neither the host nor controller secret configured, and no authentication is performed.</li> <li>• unidirectional: The host has a host secret configured. The controller will authenticate the host.</li> <li>• bidirectional: The host has both a host and controller secret configured. The controller will authenticate the host and the host will authenticate the controller.</li> </ul>

tls

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

Name	Type	Description
configured_psk	string	<p>A user supplied pre-shared key (PSK) value in PSK Interchange Format. Optional in POST.</p> <p>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>.</p> <p>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.</p>

Name	Type	Description
key_type	string	<p>The method by which the TLS pre-shared key (PSK) is configured for the host. Optional in POST.</p> <p>The values for property <code>key_type</code> and property <code>configured_psk</code> must logically agree.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• <code>none</code> - TLS is not configured for the host connection. No value is allowed for property <code>configured_psk</code>.</li> <li>• <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.</li> </ul> <p>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>.</p>

## hosts

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

Name	Type	Description
dh_hmac_chap	<a href="#">consistency_group_nvme_host_dh_hmac_chap</a>	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 io_queue count and I/O queue depth are being used.
tls	<a href="#">tls</a>	A container for the configuration for NVMe/TCP-TLS transport session for the host.

### subsystem

The NVMe subsystem to which the NVMe namespace is mapped.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
comment	string	A configurable comment for the NVMe subsystem. Optional in POST.
hosts	array[ <a href="#">hosts</a> ]	The NVMe hosts configured for access to the NVMe subsystem. Optional in POST.
name	string	The name of the NVMe subsystem. Once created, an NVMe subsystem cannot be renamed. Optional in POST.
os_type	string	The host operating system of the NVMe subsystem's hosts. Optional in POST.
uuid	string	The unique identifier of the NVMe subsystem.

### subsystem\_map

The NVMe subsystem with which the NVMe namespace is associated. A namespace can be mapped to zero (0) or one (1) subsystems.

There is an added computational cost to retrieving property values for `subsystem_map`. They are not populated for a GET request unless explicitly requested using the `fields` query parameter. See [Requesting specific fields](#) to learn more. These properties are supported only for GET.

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

svm

SVM, applies only to SVM-scoped objects.

Name	Type	Description
<a href="#">_links</a>	<a href="#">_links</a>	
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\_namespace

An NVMe namespace is a collection of addressable logical blocks presented to hosts connected to the storage virtual machine using the NVMe over Fabrics protocol.

An NVMe namespace must be mapped to an NVMe subsystem to grant access to the subsystem's hosts. Hosts can then access the NVMe namespace and perform I/O using the NVMe over Fabrics protocol.

See the NVMe namespace object model to learn more about each of the properties supported by the NVMe namespace REST API. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.

NVMe namespace names are paths of the form `"/vol/<volume>[/<qtree>]/<namespace>"` where the qtree name is optional.

An NVMe namespace is created to a specified size using thin or thick provisioning as determined by the volume on which it is created. An NVMe namespace can then be resized or cloned. An NVMe namespace cannot be renamed, or moved to a different volume. NVMe namespaces do not support the assignment of a QoS policy for performance management, but a QoS policy can be assigned to the volume containing the namespace.

`</namespace></qtree></volume>`

Name	Type	Description
<code>_links</code>	<a href="#">_links</a>	
<code>auto_delete</code>	boolean	<p>This property marks the NVMe namespace for auto deletion when the volume containing the namespace runs out of space. This is most commonly set on namespace clones.</p> <p>When set to <i>true</i>, the NVMe namespace becomes eligible for automatic deletion when the volume runs out of space. Auto deletion only occurs when the volume containing the namespace is also configured for auto deletion and free space in the volume decreases below a particular threshold.</p> <p>This property is optional in POST and PATCH. The default value for a new NVMe namespace is <i>false</i>.</p> <p>There is an added computational cost to retrieving this property's value. It is not populated for a GET request unless it is explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more.</p>

Name	Type	Description
clone	<a href="#">clone</a>	<p>This sub-object is used in POST to create a new NVMe namespace as a clone of an existing namespace, or PATCH to overwrite an existing namespace as a clone of another. Setting a property in this sub-object indicates that a namespace clone is desired.</p> <p>When used in a PATCH, the patched NVMe namespace's data is over-written as a clone of the source and the following properties are preserved from the patched namespace unless otherwise specified as part of the PATCH: <code>auto_delete</code> (unless specified in the request), <code>subsystem_map</code>, <code>status.state</code>, and <code>uuid</code>.</p>
comment	string	A configurable comment available for use by the administrator. Valid in POST and PATCH.
consistency_group	<a href="#">consistency_group</a>	The namespace's consistency group. This property is populated for namespaces that are members of a consistency group. If the namespace is a member of a child consistency group, the parent consistency group is reported. A namespace's consistency group is the consistency group of its containing volume.
convert	<a href="#">convert</a>	This sub-object is used in POST to convert a valid in-place LUN to an NVMe namespace. Setting a property in this sub-object indicates that a conversion from the specified LUN to NVMe namespace is desired.
create_time	string	The time the NVMe namespace was created.

Name	Type	Description
enabled	boolean	The enabled state of the NVMe namespace. Certain error conditions cause the namespace to become disabled. If the namespace is disabled, check the <code>status.state</code> property to determine what error disabled the namespace. An NVMe namespace is enabled automatically when it is created.
location	<a href="#">location</a>	<p>The location of the NVMe namespace within the ONTAP cluster. NVMe namespaces do not support rename, or movement between volumes. Valid in POST.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.6</li> <li>• readCreate: 1</li> </ul>
metric	<a href="#">metric</a>	Performance numbers, such as IOPS latency and throughput
name	string	<p>The name of the NVMe namespace. An NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.</p> <p>NVMe namespace names are paths of the form <code>"/vol/&lt;volume&gt;[/&lt;qtree&gt;]/&lt;namespace&gt;"</code> where the qtree name is optional.</p> <p>Renaming an NVMe namespace is not supported. Valid in POST.</p>
os_type	string	<p>The operating system type of the NVMe namespace.</p> <p>Required in POST when creating an NVMe namespace that is not a clone of another. Disallowed in POST when creating a namespace clone.</p>

Name	Type	Description
provisioning_options	<a href="#">provisioning_options</a>	Options that are applied to the operation.
space	<a href="#">space</a>	The storage space related properties of the NVMe namespace.
statistics	<a href="#">statistics</a>	These are raw performance numbers, such as IOPS latency and throughput. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.
status	<a href="#">status</a>	Status information about the NVMe namespace.
subsystem_map	<a href="#">subsystem_map</a>	<p>The NVMe subsystem with which the NVMe namespace is associated. A namespace can be mapped to zero (0) or one (1) subsystems.</p> <p>There is an added computational cost to retrieving property values for <code>subsystem_map</code>. They are not populated for a GET request unless explicitly requested using the <code>fields</code> query parameter. See <a href="#">Requesting specific fields</a> to learn more. These properties are supported only for GET.</p>
svm	<a href="#">svm</a>	SVM, applies only to SVM-scoped objects.
uuid	string	The unique identifier of the NVMe namespace.

#### error\_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

#### returned\_error

Name	Type	Description
arguments	array[ <a href="#">error_arguments</a> ]	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.