



NVMe

ONTAP 9.7 REST API reference

NetApp
April 02, 2024

Table of Contents

- NVMe 1
 - NVMe overview 1
 - View NVMe interfaces 2
 - Manage NVMe services 18
 - View NVMe subsystem controllers 80
 - Manage NVMe subsystem maps 102
 - Manage NVMe subsystems 136
 - Manage NVMe namespaces 205

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

Retrieves NVMe interfaces.

Related ONTAP commands

- `vserver nvme show-interface`

Learn more

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

Parameters

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

Name	Type	In	Required	Description
transport_address	string	query	False	Filter by transport_address
svm.uuid	string	query	False	Filter by svm.uuid
svm.name	string	query	False	Filter by svm.name
uuid	string	query	False	Filter by uuid
node.uuid	string	query	False	Filter by node.uuid
node.name	string	query	False	Filter by node.name
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.name	string	query	False	Filter by fc_interface.port.name
fc_interface.port.node.name	string	query	False	Filter by fc_interface.port.node.name
fc_interface.port.uuid	string	query	False	Filter by fc_interface.port.uuid
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.

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

Response

Status: 200, Ok

Name	Type	Description
_links	_links	
num_records	integer	Number of records.
records	array[nvme_interface]	

Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "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"
    },
    "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",
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}

```

Error

Status: Default, Error

Name	Type	Description
error	error	

Example error

```

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

```

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

_links

Name	Type	Description
self	href	

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	_links	
name	string	The name of the FC port.
node	node	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.

Name	Type	Description
_links	_links	
port	port	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.

node

Name	Type	Description
_links	_links	
name	string	
uuid	string	

svm

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

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	_links	

Name	Type	Description
enabled	boolean	The administrative state of the NVMe interface.
fc_interface	fc_interface	The attributes specific to a Fibre Channel-based NVMe interface.
name	string	The name of the NVMe interface.
node	node	
svm	svm	
transport_address	string	The transport address of 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

error

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

Retrieve an NVMe interface

GET /protocols/nvme/interfaces/{uuid}

Retrieves an NVMe interface.

Related ONTAP commands

- `vserver nvme show-interface`

Learn more

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

Parameters

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

Response

Status: 200, Ok

Name	Type	Description
_links	_links	
enabled	boolean	The administrative state of the NVMe interface.
fc_interface	fc_interface	The attributes specific to a Fibre Channel-based NVMe interface.
name	string	The name of the NVMe interface.
node	node	
svm	svm	
transport_address	string	The transport address of 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"
  },
  "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",
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
```

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
2621462	The supplied SVM does not exist.

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

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	_links	
name	string	The name of the FC port.
node	node	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.

Name	Type	Description
_links	_links	
port	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
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.

node

Name	Type	Description
_links	_links	
name	string	
uuid	string	

svm

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

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

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/bfb1beb0-
dc69-11e8-b29f-005056bb7341' -H 'accept: application/hal+json'
```

Retrieve NVMe services

GET /protocols/nvme/services

Retrieves NVMe services.

Expensive properties

There is an added cost to retrieving values for these properties. They are not included by default in GET results and must be explicitly requested using the `fields` query parameter. See [DOC 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
svm.uuid	string	query	False	Filter by svm.uuid
svm.name	string	query	False	Filter by svm.name
metric.duration	string	query	False	Filter by metric.duration
metric.latency.other	integer	query	False	Filter by metric.latency.other
metric.latency.write	integer	query	False	Filter by metric.latency.write

Name	Type	In	Required	Description
metric.latency.read	integer	query	False	Filter by metric.latency.read
metric.latency.total	integer	query	False	Filter by metric.latency.total
metric.throughput.read	integer	query	False	Filter by metric.throughput.read
metric.throughput.write	integer	query	False	Filter by metric.throughput.write
metric.throughput.total	integer	query	False	Filter by metric.throughput.total
metric.timestamp	string	query	False	Filter by metric.timestamp
metric.iops.other	integer	query	False	Filter by metric.iops.other
metric.iops.write	integer	query	False	Filter by metric.iops.write
metric.iops.read	integer	query	False	Filter by metric.iops.read
metric.iops.total	integer	query	False	Filter by metric.iops.total
metric.status	string	query	False	Filter by metric.status
statistics.status	string	query	False	Filter by statistics.status
statistics.latency_raw.other	integer	query	False	Filter by statistics.latency_raw.other
statistics.latency_raw.write	integer	query	False	Filter by statistics.latency_raw.write

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

Name	Type	In	Required	Description
return_records	boolean	query	False	The default is true for GET calls. When set to false, only the number of records is returned.
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.
order_by	array[string]	query	False	Order results by specified fields and optional [asc

Response

Status: 200, Ok

Name	Type	Description
_links	_links	
num_records	integer	Number of records.
records	array[nvme_service]	

Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "records": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "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 11:20:13 UTC"
    },
    "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 11:20:13 UTC"
},
"svm": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "svm1",
  "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
}
}
}

```

Error

Status: Default, Error

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

_links

Name	Type	Description
self	href	

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.

metric

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	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.
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.

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

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.
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.
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.

svm

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

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
_links	_links	
enabled	boolean	The administrative state of the NVMe service. The NVMe service can be disabled to block all NVMe connectivity to the SVM. This is optional in POST and PATCH. The default setting is <i>true</i> (enabled) in POST.
metric	metric	
statistics	statistics	
svm	svm	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

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

Create an NVMe service

POST `/protocols/nvme/services`

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	The default is false. If set to true, the records are returned.

Request Body

Name	Type	Description
<code>_links</code>	_links	
enabled	boolean	The administrative state of the NVMe service. The NVMe service can be disabled to block all NVMe connectivity to the SVM. This is optional in POST and PATCH. The default setting is <i>true</i> (enabled) in POST.
metric	metric	

Name	Type	Description
statistics	statistics	
svm	svm	

Example request

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "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 11:20:13 UTC"
},
"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 11:20:13 UTC"
},
"svm": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "svm1",
  "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
}
}

```

Response

Status: 201, Created

Name	Type	Description
_links	_links	
num_records	integer	Number of records.
records	array[nvme_service]	

Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "records": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "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 11:20:13 UTC"
    },
    "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 11:20:13 UTC"
},
"svm": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "svm1",
  "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
}
}
}

```

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.
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.

Error Code	Description
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.

Name	Type	Description
error	error	

Example error

```

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

```

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

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.

metric

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.
latency	latency	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.

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.

Name	Type	Description
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

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

svm

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

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
_links	_links	
enabled	boolean	The administrative state of the NVMe service. The NVMe service can be disabled to block all NVMe connectivity to the SVM. This is optional in POST and PATCH. The default setting is <i>true</i> (enabled) in POST.
metric	metric	
statistics	statistics	
svm	svm	

_links

Name	Type	Description
next	href	
self	href	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Delete an NVMe service

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

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	

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.

Error Code	Description
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.

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Retrieve an NVMe service

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

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	_links	
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	metric	
statistics	statistics	
svm	svm	

Example response

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "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 11:20:13 UTC"
},
"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 11:20:13 UTC"
},
"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.

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

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.

metric

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.
latency	latency	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.

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.

Name	Type	Description
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

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.
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.
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.

svm

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

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Update an NVMe service

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

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	_links	
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	metric	
statistics	statistics	
svm	svm	

Example request

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "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 11:20:13 UTC"
},
"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 11:20:13 UTC"
},
"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.

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

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.

metric

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.
latency	latency	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.

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.

Name	Type	Description
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

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.
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.
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.

svm

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

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
_links	_links	
enabled	boolean	The administrative state of the NVMe service. The NVMe service can be disabled to block all NVMe connectivity to the SVM. This is optional in POST and PATCH. The default setting is <i>true</i> (enabled) in POST.
metric	metric	
statistics	statistics	
svm	svm	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

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

Retrieve NVMe protocol historical performance metrics

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

Retrieves historical performance metrics for NVMe protocol of an SVM.

Parameters

Name	Type	In	Required	Description
throughput.other	integer	query	False	Filter by throughput.other
throughput.write	integer	query	False	Filter by throughput.write
throughput.read	integer	query	False	Filter by throughput.read
throughput.total	integer	query	False	Filter by throughput.total
duration	string	query	False	Filter by duration
latency.other	integer	query	False	Filter by latency.other
latency.write	integer	query	False	Filter by latency.write
latency.read	integer	query	False	Filter by latency.read
latency.total	integer	query	False	Filter by latency.total
status	string	query	False	Filter by status
timestamp	string	query	False	Filter by timestamp
iops.other	integer	query	False	Filter by iops.other
iops.write	integer	query	False	Filter by iops.write

Name	Type	In	Required	Description
iops.read	integer	query	False	Filter by iops.read
iops.total	integer	query	False	Filter by iops.total
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"> • Default value: 1
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
<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"> • Default value: 1 	svm.uuid	string	path	True
Unique identifier of the SVM.	interval	string	query	False

Response

Status: 200, Ok

Name	Type	Description
_links	_links	
num_records	integer	Number of records
records	array[records]	

Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "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 11:20:13 UTC"
  }
}
```

Error

Status: Default, Error

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

_links

Name	Type	Description
self	href	

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
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.

records

Performance numbers, such as IOPS latency and throughput.

Name	Type	Description
_links	_links	

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	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	Errors associated with the sample. For example, if the aggregation of data over multiple nodes fails, then any partial errors might return "ok" on success or "error" on an 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.

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

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_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",
"_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/{subsystem.uuid}

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

# The response:
{
  "records": [
    {
      "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_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_fc_nvme_symmcon_fc_nvme_vserver_0_3a_0",
    "uuid": "fal5941-2593-11e9-94c4-00a0989a1c8e",
    "transport_address": "nn-0x200400a0989a1c8d:pn-0x200500a0989a1c8d",
    "_links": {
      "self": {
        "href": "/api/protocols/nvme/interfaces/fal5941-2593-11e9-94c4-00a0989a1c8e"
      }
    }
  },
  "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
  },
  "_links": {
    "self": {
      "href": "/api/protocols/nvme/subsystem-controllers/14875240-2594-11e9-abde-00a098984313/0040h"
    }
  }
}

```



```
}  
}  
}
```

Retrieve NVMe subsystem controllers

GET /protocols/nvme/subsystem-controllers

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
io_queue.count	integer	query	False	Filter by io_queue.count
io_queue.depth	integer	query	False	Filter by io_queue.depth
admin_queue.depth	integer	query	False	Filter by admin_queue.depth
id	string	query	False	Filter by id
host.id	string	query	False	Filter by host.id
host.nqn	string	query	False	Filter by host.nqn
host.transport_address	string	query	False	Filter by host.transport_address
subsystem.uuid	string	query	False	Filter by subsystem.uuid
subsystem.name	string	query	False	Filter by subsystem.name
node.uuid	string	query	False	Filter by node.uuid

Name	Type	In	Required	Description
node.name	string	query	False	Filter by node.name
interface.uuid	string	query	False	Filter by interface.uuid
interface.transport_address	string	query	False	Filter by interface.transport_address
interface.name	string	query	False	Filter by interface.name
svm.uuid	string	query	False	Filter by svm.uuid
svm.name	string	query	False	Filter by svm.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	The default is true for GET calls. When set to false, only the number of records is returned.
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.
order_by	array[string]	query	False	Order results by specified fields and optional [asc

Response

Status: 200, Ok

Name	Type	Description
_links	_links	
num_records	integer	Number of records.
records	array[nvme_subsystem_controller]	

Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "records": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "admin_queue": {
      "depth": 0
    },
    "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": "falc5941-2593-11e9-94c4-00a0989a1c8e"
    },
    "io_queue": {
      "count": 0,
      "depth": {
      }
    },
    "node": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "node1",

```

```

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

```

Error

Status: Default, Error

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

_links

Name	Type	Description
self	href	

admin_queue

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

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.

Name	Type	Description
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	_links	
name	string	
uuid	string	

subsystem

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

svm

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

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	_links	
admin_queue	admin_queue	
host	host	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	interface	The logical interface through which the host is connected.
io_queue	io_queue	Properties of the I/O queues available to the controller.
node	node	
subsystem	subsystem	
svm	svm	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

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

Retrieve an NVMe subsystem controller

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

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
_links	_links	
admin_queue	admin_queue	
host	host	Properties of the connected host.

Name	Type	Description
id	string	The identifier of the subsystem controller. This field consists of 4 zero-filled hexadecimal digits followed by an 'h'.
interface	interface	The logical interface through which the host is connected.
io_queue	io_queue	Properties of the I/O queues available to the controller.
node	node	
subsystem	subsystem	
svm	svm	

Example response

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "admin_queue": {
    "depth": 0
  },
  "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": {
    }
  },
  "node": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "node1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "subsystem": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "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
72090001	The supplied subsystem identifier does not exist.

Name	Type	Description
error	error	

Example error

```

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

```

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

admin_queue

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

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	_links	
name	string	
uuid	string	

subsystem

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

svm

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

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

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/namespacel",
    "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
```

Retrieves NVMe subsystem maps.

Expensive properties

There is an added cost to retrieving values for these properties. They are not included by default in GET results and must be explicitly requested using the `fields` query parameter. See [DOC 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
<code>nsid</code>	string	query	False	Filter by nsid
<code>subsystem.uuid</code>	string	query	False	Filter by subsystem.uuid
<code>subsystem.name</code>	string	query	False	Filter by subsystem.name
<code>anagrpid</code>	string	query	False	Filter by anagrpid
<code>svm.uuid</code>	string	query	False	Filter by svm.uuid
<code>svm.name</code>	string	query	False	Filter by svm.name
<code>namespace.uuid</code>	string	query	False	Filter by namespace.uuid
<code>namespace.node.uuid</code>	string	query	False	Filter by namespace.node.uuid
<code>namespace.node.name</code>	string	query	False	Filter by namespace.node.name
<code>namespace.name</code>	string	query	False	Filter by namespace.name
<code>fields</code>	array[string]	query	False	Specify the fields to return.
<code>max_records</code>	integer	query	False	Limit the number of records returned.

Name	Type	In	Required	Description
return_records	boolean	query	False	The default is true for GET calls. When set to false, only the number of records is returned.
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.
order_by	array[string]	query	False	Order results by specified fields and optional [asc

Response

Status: 200, Ok

Name	Type	Description
_links	_links	
num_records	integer	Number of records.
records	array[nvme_subsystem_map]	

Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "records": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "anagrpId": "00103050h",
    "namespace": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "/vol/vol1/namespacel",
      "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"
        }
      },
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "svm": {
```

```
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "svm1",
  "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
}
}
```

Error

Status: Default, Error

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

_links

Name	Type	Description
self	href	

node

Name	Type	Description
_links	_links	
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	_links	
name	string	The fully qualified path name of the NVMe namespace composed from the volume name, qtree name, and file name of the NVMe namespace. Valid in POST.
node	node	
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	_links	
name	string	The name of the NVMe subsystem.
uuid	string	The unique identifier of the NVMe subsystem.

svm

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

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	_links	
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 cost to retrieving this property's value. It is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>

Name	Type	Description
namespace	namespace	The NVMe namespace to which the NVMe subsystem is mapped. Required in POST by supplying either the UUID, name, or both.
nsid	string	The NVMe namespace identifier. This is an identifier used by an NVMe controller to provide access to the NVMe namespace. The format for an NVMe namespace identifier is 8 hexadecimal digits (zero-filled) followed by a lower case "h".
subsystem	subsystem	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	svm	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Create an NVMe subsystem map

POST /protocols/nvme/subsystem-maps

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 create`

Learn more

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

Parameters

Name	Type	In	Required	Description
<code>return_records</code>	boolean	query	False	The default is false. If set to true, the records are returned.

Request Body

Name	Type	Description
<code>_links</code>	_links	

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 cost to retrieving this property's value. It is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>
namespace	namespace	<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	subsystem	<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	svm	

Example request

A large, empty rectangular box with a thin, dashed border, occupying most of the page. It is intended for an 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"
      }
    },
    "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	_links	
num_records	integer	Number of records.
records	array[nvme_subsystem_map]	

Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "records": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "anagrpId": "00103050h",
    "namespace": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "/vol/vol1/namespacel",
      "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"
        }
      },
      "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
72089790	The supplied NVMe namespace is already mapped to the supplied NVMe subsystem.
72089793	An NVMe namespace in a Snapshot copy 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.
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.

Error Code	Description
72090021	The NVMe subsystem specified by <code>subsystem.name</code> was not found.

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

node

Name	Type	Description
_links	_links	
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	_links	
name	string	The fully qualified path name of the NVMe namespace composed from the volume name, qtree name, and file name of the NVMe namespace. Valid in POST.
node	node	
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	_links	

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

svm

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

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	_links	
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 cost to retrieving this property's value. It is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>
namespace	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
nsid	string	The NVMe namespace identifier. This is an identifier used by an NVMe controller to provide access to the NVMe namespace. The format for an NVMe namespace identifier is 8 hexadecimal digits (zero-filled) followed by a lower case "h".
subsystem	subsystem	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	svm	

_links

Name	Type	Description
next	href	
self	href	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Delete an NVMe subsystem map

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

Deletes an NVMe subsystem map.

Related ONTAP commands

- `vserver nvme subsystem map delete`

Learn more

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

Parameters

Name	Type	In	Required	Description
subsystem.uuid	string	path	True	
namespace.uuid	string	path	True	

Response

```
Status: 200, Ok
```

Name	Type	Description
_links	_links	
num_records	integer	Number of records.
records	array[nvme_subsystem_map]	

Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "records": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "anagrpid": "00103050h",
    "namespace": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "/vol/vol1/namespacel",
      "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"
        }
      },
      "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.

Name	Type	Description
error	error	

Example error

```

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

```

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

_links

Name	Type	Description
self	href	

node

Name	Type	Description
_links	_links	
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	_links	
name	string	The fully qualified path name of the NVMe namespace composed from the volume name, qtree name, and file name of the NVMe namespace. Valid in POST.
node	node	
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	_links	
name	string	The name of the NVMe subsystem.
uuid	string	The unique identifier of the NVMe subsystem.

svm

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

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	_links	
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 cost to retrieving this property's value. It is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>

Name	Type	Description
namespace	namespace	The NVMe namespace to which the NVMe subsystem is mapped. Required in POST by supplying either the UUID, name, or both.
nsid	string	The NVMe namespace identifier. This is an identifier used by an NVMe controller to provide access to the NVMe namespace. The format for an NVMe namespace identifier is 8 hexadecimal digits (zero-filled) followed by a lower case "h".
subsystem	subsystem	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	svm	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Retrieve an NVMe subsystem map

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

Retrieves an NVMe subsystem map.

Expensive properties

There is an added cost to retrieving values for these properties. They are not included by default in GET results and must be explicitly requested using the `fields` query parameter. See [DOC 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.
fields	array[string]	query	False	Specify the fields to return.

Response

Status: 200, Ok

Name	Type	Description
<code>_links</code>	_links	

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 cost to retrieving this property's value. It is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>
namespace	namespace	<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	subsystem	<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	svm	

Example response

A large, empty rectangular box with a thin, dashed border, occupying most of the page. It is intended for an 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"
      }
    },
    "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.

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

node

Name	Type	Description
_links	_links	
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	_links	
name	string	The fully qualified path name of the NVMe namespace composed from the volume name, qtree name, and file name of the NVMe namespace. Valid in POST.
node	node	
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	_links	

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

svm

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

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

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",
  "io_queue": {
    "count": 4,
    "depth": 32
  },
}
```

Deleting an NVMe subsystem host from an NVMe subsystem

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

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

Retrieve NVMe subsystems

```
GET /protocols/nvme/subsystems
```

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
target_nqn	string	query	False	Filter by target_nqn
uuid	string	query	False	Filter by uuid
subsystem_maps.nsid	string	query	False	Filter by subsystem_maps.nsid
subsystem_maps.ana_grpid	string	query	False	Filter by subsystem_maps.ana_grpid
subsystem_maps.namespace.uuid	string	query	False	Filter by subsystem_maps.namespace.uuid
subsystem_maps.namespace.name	string	query	False	Filter by subsystem_maps.namespace.name
serial_number	string	query	False	Filter by serial_number
os_type	string	query	False	Filter by os_type
svm.uuid	string	query	False	Filter by svm.uuid
svm.name	string	query	False	Filter by svm.name
comment	string	query	False	Filter by comment
delete_on_unmap	boolean	query	False	Filter by delete_on_unmap
io_queue.default.depth	integer	query	False	Filter by io_queue.default.depth

Name	Type	In	Required	Description
io_queue.default.count	integer	query	False	Filter by io_queue.default.count
name	string	query	False	Filter by name
hosts.nqn	string	query	False	Filter by hosts.nqn
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.
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.
order_by	array[string]	query	False	Order results by specified fields and optional [asc

Response

Status: 200, Ok

Name	Type	Description
_links	_links	

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

Example response

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

Error

Status: Default, Error

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

_links

Name	Type	Description
self	href	

hosts

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

default

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

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

io_queue

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

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

namespace

An NVMe namespace mapped to the NVMe subsystem.

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

subsystem_maps

An NVMe namespace mapped to the NVMe subsystem.

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

svm

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

nvme_subsystem

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

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

Name	Type	Description
svm	svm	
target_nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target.
uuid	string	The unique identifier of the NVMe subsystem.

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Create an NVMe subsystem

POST `/protocols/nvme/subsystems`

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.

Request Body

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

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

Example request

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

Response

Status: 201, Created

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

Example response

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

```

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.
72089709	The NVMe subsystem name contains an invalid character.
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> .
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.

Error Code	Description
72090035	Passing NVMe subsystem host NQNs on NVMe subsystem POST requires an effective cluster version of 9.7 or later.
72090036	The <code>hosts.nqn</code> NVMe subsystem property must contain unique values.

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

hosts

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

default

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

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

io_queue

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

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

namespace

An NVMe namespace mapped to the NVMe subsystem.

Name	Type	Description
_links	_links	

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

subsystem_maps

An NVMe namespace mapped to the NVMe subsystem.

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

svm

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

nvme_subsystem

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

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

_links

Name	Type	Description
next	href	
self	href	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Retrieve NVMe subsystem hosts

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

Retrieves the NVMe subsystem hosts of an NVMe subsystem.

Expensive properties

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

- `subsystem_maps.*`

Related ONTAP commands

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

Learn more

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

Parameters

Name	Type	In	Required	Description
subsystem.uuid	string	path	True	The unique identifier of the NVMe subsystem.
fields	array[string]	query	False	Specify the fields to return.
max_records	integer	query	False	Limit the number of records returned.
return_records	boolean	query	False	The default is true for GET calls. When set to false, only the number of records is returned.
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.
order_by	array[string]	query	False	Order results by specified fields and optional [asc

Response

Status: 200, Ok

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

Example response

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

```
    }
  },
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
}
```

Error

Status: Default, Error

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

_links

Name	Type	Description
self	href	

io_queue

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

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

subsystem

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

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

records

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

Name	Type	Description
_links	_links	
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.
nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target. Not allowed in POST when the <code>records</code> property is used.
subsystem	subsystem	The NVMe subsystem to which the NVMe host has been provisioned.

nvme_subsystem_host

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

Name	Type	Description
_links	_links	
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.
nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target. Not allowed in POST when the <code>records</code> property is used.
records	array[records]	An array of NVMe hosts specified to add multiple NVMe hosts to an NVMe subsystem in a single API call. Valid in POST only.
subsystem	subsystem	The NVMe subsystem to which the NVMe host has been provisioned.

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Add NVMe subsystem hosts

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

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
return_records	boolean	query	False	The default is false. If set to true, the records are returned.

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

Request Body

Name	Type	Description
_links	_links	
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.
nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target. Not allowed in POST when the <code>records</code> property is used.
records	array[records]	An array of NVMe hosts specified to add multiple NVMe hosts to an NVMe subsystem in a single API call. Valid in POST only.
subsystem	subsystem	The NVMe subsystem to which the NVMe host has been provisioned.

Example request

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


Response

Status: 201, Created

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

Example response

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

```

    }
  },
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
}
}

```

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
72089705	The NVMe subsystem host already exists for the NVMe subsystem.
72089771	The NQN is invalid. A non-empty qualifier is required after the prefix. An example of a valid NQN is <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> .
72090002	The POST request of hosts to an NVMe subsystem can only contain an 'nqn' property or 'records' property, but not both.
72090003	The elements in the records array for a POST of hosts to an NVMe subsystem must contain only the nqn property.

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

io_queue

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

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

subsystem

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

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

records

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

Name	Type	Description
_links	_links	

Name	Type	Description
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.
nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target. Not allowed in POST when the <code>records</code> property is used.
subsystem	subsystem	The NVMe subsystem to which the NVMe host has been provisioned.

nvme_subsystem_host

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

Name	Type	Description
_links	_links	
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.
nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target. Not allowed in POST when the <code>records</code> property is used.
records	array[records]	An array of NVMe hosts specified to add multiple NVMe hosts to an NVMe subsystem in a single API call. Valid in POST only.
subsystem	subsystem	The NVMe subsystem to which the NVMe host has been provisioned.

_links

Name	Type	Description
next	href	

Name	Type	Description
self	href	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Delete an NVMe subsystem host

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

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	
nqn	string	path	True	

Response

Status: 200, Ok

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
72089771	The NQN is invalid. A non-empty qualifier is required after the prefix. An example of a valid NQN is <i>nqn.1992-01.com.example:string</i> .
72089772	The NQN is invalid. Add the prefix 'nqn'. An example of a valid NQN is <i>nqn.1992-01.com.example:string</i> .
72089773	The NQN is invalid. The date field must be formatted <i>yyyy-mm</i> . An example of a valid NQN is <i>nqn.1992-01.com.example:string</i> .

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Retrieve an NVMe subsystem host

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

Retrieves an NVMe subsystem host of an NVMe subsystem.

Related ONTAP commands

- `vserver nvme subsystem host show`

Learn more

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

Parameters

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

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

Response

Status: 200, Ok

Name	Type	Description
_links	_links	
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.
nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target. Not allowed in POST when the <code>records</code> property is used.
records	array[records]	An array of NVMe hosts specified to add multiple NVMe hosts to an NVMe subsystem in a single API call. Valid in POST only.
subsystem	subsystem	The NVMe subsystem to which the NVMe host has been provisioned.

Example response

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

Error

Status: Default, Error

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

io_queue

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

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

subsystem

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

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

records

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

Name	Type	Description
_links	_links	

Name	Type	Description
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.
nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target. Not allowed in POST when the <code>records</code> property is used.
subsystem	subsystem	The NVMe subsystem to which the NVMe host has been provisioned.

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Remove an NVMe subsystem

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

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	
allow_delete_while_mapped	boolean	query	False	
allow_delete_with_hosts	boolean	query	False	

Response

Status: 200, Ok

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
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.

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Retrieve an NVMe subsystem

GET /protocols/nvme/subsystems/{uuid}

Retrieves an NVMe subsystem.

Expensive properties

There is an added cost to retrieving values for these properties. They are not included by default in GET results and must be explicitly requested using the `fields` query parameter. See [DOC 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
<code>uuid</code>	<code>string</code>	<code>path</code>	True	The unique identifier of the NVMe subsystem.
<code>fields</code>	<code>array[string]</code>	<code>query</code>	False	Specify the fields to return.

Response

Status: 200, Ok

Name	Type	Description
<code>_links</code>	_links	
<code>comment</code>	<code>string</code>	A configurable comment for the NVMe subsystem. Optional in POST and PATCH.
<code>delete_on_unmap</code>	<code>boolean</code>	An option that causes the subsystem to be deleted when the last subsystem map associated with it is deleted. This property defaults to <i>false</i> when the subsystem is created.
<code>hosts</code>	<code>array[hosts]</code>	

Name	Type	Description
io_queue	io_queue	The properties of the submission queue used to submit I/O commands for execution by the NVMe controller.
name	string	The name of the NVMe subsystem. Once created, an NVMe subsystem cannot be renamed. Required in POST.
os_type	string	The host operating system of the NVMe subsystem's hosts. Required in POST.
serial_number	string	The serial number of the NVMe subsystem.
subsystem_maps	array[subsystem_maps]	<p>The NVMe namespaces mapped to the NVMe subsystem.</p> <p>There is an added cost to retrieving property values for <code>subsystem_maps</code>. They are not populated for either a collection GET or an instance GET unless explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>
svm	svm	
target_nqn	string	The NVMe qualified name (NQN) used to identify the NVMe storage target.
uuid	string	The unique identifier of the NVMe subsystem.

Example response

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

Error

Status: Default, Error

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

hosts

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

default

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

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

io_queue

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

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

namespace

An NVMe namespace mapped to the NVMe subsystem.

Name	Type	Description
_links	_links	

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

subsystem_maps

An NVMe namespace mapped to the NVMe subsystem.

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

svm

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

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Update an NVMe subsystem

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

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
<code>_links</code>	_links	

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

Example request

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

Response

Status: 200, Ok

Error

Status: Default, Error

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

hosts

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

default

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

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

io_queue

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

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

namespace

An NVMe namespace mapped to the NVMe subsystem.

Name	Type	Description
_links	_links	

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

subsystem_maps

An NVMe namespace mapped to the NVMe subsystem.

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

svm

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

nvme_subsystem

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

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

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

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.

In ONTAP, an NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.

An NVMe namespace is created to a specified size using thin or thick provisioning as determined by the volume on which it is created. NVMe namespaces support being cloned. An NVMe namespace cannot be renamed, resized, 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. See the NVMe namespace object model to learn more about each of the properties supported by the NVMe namespace REST API.

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.

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/namespacel" }'

# 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/namespacel",
      "location": {
        "namespace": "namespacel",
        "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

This example sets the `comment` property 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 '{ "comment":
"Data for the research department." }'

```

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"
      }
    },
    "_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 copy, 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 copy. 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" } } }'
```

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

Retrieves NVMe namespaces.

Expensive properties

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

- `auto_delete`
- `subsystem_map.*`
- `status.mapped`

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
name	string	query	False	Filter by name
comment	string	query	False	Filter by comment
subsystem_map.subsystem.uuid	string	query	False	Filter by subsystem_map.subsystem.uuid
subsystem_map.subsystem.name	string	query	False	Filter by subsystem_map.subsystem.name
subsystem_map.nsid	string	query	False	Filter by subsystem_map.nsid
subsystem_map.anagrpId	string	query	False	Filter by subsystem_map.anagrpId

Name	Type	In	Required	Description
space.block_size	integer	query	False	Filter by space.block_size
space.used	integer	query	False	Filter by space.used
space.guarantee.requested	boolean	query	False	Filter by space.guarantee.requested
space.guarantee.reserved	boolean	query	False	Filter by space.guarantee.reserved
space.size	integer	query	False	Filter by space.size
status.mapped	boolean	query	False	Filter by status.mapped
status.state	string	query	False	Filter by status.state
status.container_state	string	query	False	Filter by status.container_state
status.read_only	boolean	query	False	Filter by status.read_only
create_time	string	query	False	Filter by create_time
os_type	string	query	False	Filter by os_type
svm.uuid	string	query	False	Filter by svm.uuid
svm.name	string	query	False	Filter by svm.name
location.namespace	string	query	False	Filter by location.namespace
location.volume.uuid	string	query	False	Filter by location.volume.uuid
location.volume.name	string	query	False	Filter by location.volume.name

Name	Type	In	Required	Description
location.qtree.name	string	query	False	Filter by location.qtree.name
location.qtree.id	integer	query	False	Filter by location.qtree.id
enabled	boolean	query	False	Filter by enabled
auto_delete	boolean	query	False	Filter by auto_delete
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.
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.
order_by	array[string]	query	False	Order results by specified fields and optional [asc

Response

Status: 200, Ok

Name	Type	Description
_links	_links	
num_records	integer	Number of records.
records	array[nvme_namespace]	

Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "records": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "clone": {
      "source": {
        "name": "/vol/volume1/namespace1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      }
    },
    "comment": "string",
    "create_time": "2018-06-04 19:00:00 UTC",
    "location": {
      "namespace": "namespace1",
      "qtree": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "id": 1,
        "name": "qt1"
      },
      "volume": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "volume1",
        "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
      }
    }
  },
}
```

```
"name": "/vol/volume1/mtree1/namespace1",
"os_type": "linux",
"space": {
  "block_size": 512,
  "size": 1073741824,
  "used": 0
},
"status": {
  "container_state": "online",
  "state": "online"
},
"subsystem_map": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "anagrpid": "00103050h",
  "nsid": "00000001h",
  "subsystem": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    }
  },
  "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	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

_links

Name	Type	Description
self	href	

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	The fully qualified path name of the clone source NVMe namespace composed of a "/vol" prefix, the volume name, the (optional) qtree name and base name of the namespace. Valid in POST and PATCH.
uuid	string	The unique identifier of the clone source NVMe namespace. Valid in POST and PATCH.

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
<code>source</code>	source	<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>

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>	_links	
<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>	_links	

Name	Type	Description
name	string	The name of the volume.
uuid	string	<p>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.</p> <ul style="list-style-type: none"> example: 028baa66-41bd-11e9-81d5-00a0986138f7

location

The location of the NVMe namespace within the ONTAP cluster. Valid in POST.

NVMe namespaces do not support rename, or movement between volumes.

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>
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>

Name	Type	Description
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>

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"> • readOnly: 1

Name	Type	Description
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.</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>
guarantee	guarantee	Properties that request and report the space guarantee for the NVMe namespace.
size	integer	<p>The total provisioned size of the NVMe namespace.</p> <p>NVMe namespaces do not support resize.</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"> • example: 1073741824 • readCreate: 1

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"> • readOnly: 1

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.

Name	Type	Description
mapped	boolean	Reports if the NVMe namespace is mapped to an NVMe subsystem. There is an added cost to retrieving this property's value. It is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields 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.

subsystem

The NVMe subsystem to which the NVMe namespace is mapped.

Name	Type	Description
_links	_links	
name	string	The name of the NVMe subsystem.
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 cost to retrieving property values for `subsystem_map`. They are not populated for either a collection GET or an instance GET unless explicitly requested using the `fields` query parameter. See [DOC Requesting specific fields](#) to learn more.

Name	Type	Description
_links	_links	

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

svm

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

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.

In ONTAP, an NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.

An NVMe namespace is created to a specified size using thin or thick provisioning as determined by the volume on which it is created. NVMe namespaces support being cloned. An NVMe namespace cannot be renamed, resized, 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. See the NVMe namespace object model to learn more about each of the properties supported by the NVMe namespace REST API.

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.

Name	Type	Description
_links	_links	
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 cost to retrieving this property's value. It is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>

Name	Type	Description
clone	clone	<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.
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, you can check the <code>state</code> property to determine what error disabled the namespace. An NVMe namespace is enabled automatically when it is created.
location	location	<p>The location of the NVMe namespace within the ONTAP cluster. Valid in POST.</p> <p>NVMe namespaces do not support rename, or movement between volumes.</p> <ul style="list-style-type: none"> • <code>readCreate: 1</code>

Name	Type	Description
name	string	<p>The fully qualified path name of the NVMe namespace composed of a "/vol" prefix, the volume name, the (optional) qtree name and base name of the namespace. Valid in POST.</p> <p>NVMe namespaces do not support rename, or movement between volumes.</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>
space	space	The storage space related properties of the NVMe namespace.
status	status	Status information about the NVMe namespace.
subsystem_map	subsystem_map	<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 cost to retrieving property values for <code>subsystem_map</code>. They are not populated for either a collection GET or an instance GET unless explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>
svm	svm	
uuid	string	The unique identifier of the NVMe namespace.

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Create an NVMe namespace

POST /storage/namespaces

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*

Related ONTAP commands

- `volume file clone autodelete`

- volume file clone create
- vserver nvme namespace create

Learn more

- [DOC /storage/namespaces](#)

Parameters

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

Request Body

Name	Type	Description
_links	_links	
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 cost to retrieving this property's value. It is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>

Name	Type	Description
clone	clone	<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.
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, you can check the <code>state</code> property to determine what error disabled the namespace. An NVMe namespace is enabled automatically when it is created.
location	location	<p>The location of the NVMe namespace within the ONTAP cluster. Valid in POST.</p> <p>NVMe namespaces do not support rename, or movement between volumes.</p> <ul style="list-style-type: none"> • <code>readCreate: 1</code>

Name	Type	Description
name	string	<p>The fully qualified path name of the NVMe namespace composed of a "/vol" prefix, the volume name, the (optional) qtree name and base name of the namespace. Valid in POST.</p> <p>NVMe namespaces do not support rename, or movement between volumes.</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>
space	space	The storage space related properties of the NVMe namespace.
status	status	Status information about the NVMe namespace.
subsystem_map	subsystem_map	<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 cost to retrieving property values for <code>subsystem_map</code>. They are not populated for either a collection GET or an instance GET unless explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>
svm	svm	
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",
  "create_time": "2018-06-04 19:00:00 UTC",
  "location": {
    "namespace": "namespace1",
    "qtree": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "id": 1,
      "name": "qt1"
    },
    "volume": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "volume1",
      "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
    }
  },
  "name": "/vol/volume1/qtree1/namespace1",
  "os_type": "linux",
  "space": {
    "block_size": 512,
    "size": 1073741824,
    "used": 0
  },
  "status": {
    "container_state": "online",

```

```

    "state": "online"
  },
  "subsystem_map": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "anagrpid": "00103050h",
    "nsid": "00000001h",
    "subsystem": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "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: 201, Created

Name	Type	Description
_links	_links	
num_records	integer	Number of records.
records	array[nvme_namespace]	

Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "records": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "clone": {
      "source": {
        "name": "/vol/volume1/namespace1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      }
    },
    "comment": "string",
    "create_time": "2018-06-04 19:00:00 UTC",
    "location": {
      "namespace": "namespace1",
      "qtree": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "id": 1,
        "name": "qt1"
      },
      "volume": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "volume1",
        "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
      }
    }
  },
}
```

```
"name": "/vol/volume1/mtree1/namespace1",
"os_type": "linux",
"space": {
  "block_size": 512,
  "size": 1073741824,
  "used": 0
},
"status": {
  "container_state": "online",
  "state": "online"
},
"subsystem_map": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "anagrpid": "00103050h",
  "nsid": "00000001h",
  "subsystem": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    }
  },
  "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
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.
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 qtree was not found.
5242950	The specified <code>location.qtree.id</code> and <code>location.qtree.name</code> do not refer to the same qtree.
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.
13565952	The NVMe namespace clone request failed.
72089720	NVMe namespaces cannot be created in Snapshot copies.
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.

Error Code	Description
72089727	NVMe namespaces cannot be created on an SVM root volume.
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 copy.

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

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	The fully qualified path name of the clone source NVMe namespace composed of a "/vol" prefix, the volume name, the (optional) qtree name and base name of the namespace. Valid in POST and PATCH.
uuid	string	The unique identifier of the clone source NVMe namespace. Valid in POST and PATCH.

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	source	<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>

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
_links	_links	
id	integer	The identifier for the qtree, unique within the qtree's volume.
name	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
_links	_links	
name	string	The name of the volume.

Name	Type	Description
uuid	string	<p>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.</p> <ul style="list-style-type: none"> • example: 028baa66-41bd-11e9-81d5-00a0986138f7

location

The location of the NVMe namespace within the ONTAP cluster. Valid in POST.

NVMe namespaces do not support rename, or movement between volumes.

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>
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>

Name	Type	Description
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>

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"> • readOnly: 1

Name	Type	Description
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.</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>
guarantee	guarantee	Properties that request and report the space guarantee for the NVMe namespace.
size	integer	<p>The total provisioned size of the NVMe namespace.</p> <p>NVMe namespaces do not support resize.</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"> • example: 1073741824 • readCreate: 1

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"> • readOnly: 1

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.

Name	Type	Description
mapped	boolean	Reports if the NVMe namespace is mapped to an NVMe subsystem. There is an added cost to retrieving this property's value. It is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields 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.

subsystem

The NVMe subsystem to which the NVMe namespace is mapped.

Name	Type	Description
_links	_links	
name	string	The name of the NVMe subsystem.
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 cost to retrieving property values for `subsystem_map`. They are not populated for either a collection GET or an instance GET unless explicitly requested using the `fields` query parameter. See [DOC Requesting specific fields](#) to learn more.

Name	Type	Description
_links	_links	

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

svm

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

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.

In ONTAP, an NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.

An NVMe namespace is created to a specified size using thin or thick provisioning as determined by the volume on which it is created. NVMe namespaces support being cloned. An NVMe namespace cannot be renamed, resized, 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. See the NVMe namespace object model to learn more about each of the properties supported by the NVMe namespace REST API.

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.

Name	Type	Description
_links	_links	
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 cost to retrieving this property's value. It is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>

Name	Type	Description
clone	clone	<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.
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, you can check the <code>state</code> property to determine what error disabled the namespace. An NVMe namespace is enabled automatically when it is created.
location	location	<p>The location of the NVMe namespace within the ONTAP cluster. Valid in POST.</p> <p>NVMe namespaces do not support rename, or movement between volumes.</p> <ul style="list-style-type: none"> • <code>readCreate: 1</code>

Name	Type	Description
name	string	<p>The fully qualified path name of the NVMe namespace composed of a "/vol" prefix, the volume name, the (optional) qtree name and base name of the namespace. Valid in POST.</p> <p>NVMe namespaces do not support rename, or movement between volumes.</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>
space	space	The storage space related properties of the NVMe namespace.
status	status	Status information about the NVMe namespace.
subsystem_map	subsystem_map	<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 cost to retrieving property values for <code>subsystem_map</code>. They are not populated for either a collection GET or an instance GET unless explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>
svm	svm	
uuid	string	The unique identifier of the NVMe namespace.

[_links](#)

Name	Type	Description
next	href	
self	href	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Delete an NVMe namespace

DELETE /storage/namespaces/{uuid}

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.

Name	Type	In	Required	Description
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>This parameter should be used with caution.</p> <ul style="list-style-type: none"> • Default value:

Response

Status: 200, Ok

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
72090006	The specified namespace was not found.
72090007	The specified namespace was not found.
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.

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Retrieve an NVMe namespace

GET /storage/namespaces/{uuid}

Retrieves an NVMe namespace.

Expensive properties

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

- `auto_delete`
- `subsystem_map.*`
- `status.mapped`

Related ONTAP commands

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

Learn more

- [DOC /storage/namespaces](#)

Parameters

Name	Type	In	Required	Description
<code>uuid</code>	<code>string</code>	<code>path</code>	True	The unique identifier of the NVMe namespace to retrieve.
<code>fields</code>	<code>array[string]</code>	<code>query</code>	False	Specify the fields to return.

Response

```
Status: 200, Ok
```

Name	Type	Description
<code>_links</code>	_links	

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 cost to retrieving this property's value. It is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>
clone	clone	<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.
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, you can check the <code>state</code> property to determine what error disabled the namespace. An NVMe namespace is enabled automatically when it is created.
location	location	<p>The location of the NVMe namespace within the ONTAP cluster. Valid in POST.</p> <p>NVMe namespaces do not support rename, or movement between volumes.</p> <ul style="list-style-type: none"> • <code>readCreate: 1</code>
name	string	<p>The fully qualified path name of the NVMe namespace composed of a <code>"/vol"</code> prefix, the volume name, the (optional) <code>qtree</code> name and base name of the namespace. Valid in POST.</p> <p>NVMe namespaces do not support rename, or movement between volumes.</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
space	space	The storage space related properties of the NVMe namespace.
status	status	Status information about the NVMe namespace.
subsystem_map	subsystem_map	<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 cost to retrieving property values for <code>subsystem_map</code>. They are not populated for either a collection GET or an instance GET unless explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>
svm	svm	
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",
  "create_time": "2018-06-04 19:00:00 UTC",
  "location": {
    "namespace": "namespace1",
    "qtree": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "id": 1,
      "name": "qt1"
    },
    "volume": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "volume1",
      "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
    }
  },
  "name": "/vol/volume1/qtree1/namespace1",
  "os_type": "linux",
  "space": {
    "block_size": 512,
    "size": 1073741824,
    "used": 0
  },
  "status": {
    "container_state": "online",

```

```

    "state": "online"
  },
  "subsystem_map": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "anagrpid": "00103050h",
    "nsid": "00000001h",
    "subsystem": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "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.

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

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	The fully qualified path name of the clone source NVMe namespace composed of a "/vol" prefix, the volume name, the (optional) qtree name and base name of the namespace. Valid in POST and PATCH.
uuid	string	The unique identifier of the clone source NVMe namespace. Valid in POST and PATCH.

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	source	<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>

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
_links	_links	
id	integer	The identifier for the qtree, unique within the qtree's volume.
name	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
_links	_links	
name	string	The name of the volume.

Name	Type	Description
uuid	string	<p>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.</p> <ul style="list-style-type: none"> • example: 028baa66-41bd-11e9-81d5-00a0986138f7

location

The location of the NVMe namespace within the ONTAP cluster. Valid in POST.

NVMe namespaces do not support rename, or movement between volumes.

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>
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>

Name	Type	Description
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>

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"> • readOnly: 1

Name	Type	Description
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.</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>
guarantee	guarantee	Properties that request and report the space guarantee for the NVMe namespace.
size	integer	<p>The total provisioned size of the NVMe namespace.</p> <p>NVMe namespaces do not support resize.</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"> • example: 1073741824 • readCreate: 1

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"> • readOnly: 1

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.

Name	Type	Description
mapped	boolean	Reports if the NVMe namespace is mapped to an NVMe subsystem. There is an added cost to retrieving this property's value. It is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields 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.

subsystem

The NVMe subsystem to which the NVMe namespace is mapped.

Name	Type	Description
_links	_links	
name	string	The name of the NVMe subsystem.
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 cost to retrieving property values for `subsystem_map`. They are not populated for either a collection GET or an instance GET unless explicitly requested using the `fields` query parameter. See [DOC Requesting specific fields](#) to learn more.

Name	Type	Description
_links	_links	

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

svm

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

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

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

Update an NVMe namespace

PATCH /storage/namespaces/{uuid}

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.

Request Body

Name	Type	Description
_links	_links	

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 cost to retrieving this property's value. It is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>
clone	clone	<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.
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, you can check the <code>state</code> property to determine what error disabled the namespace. An NVMe namespace is enabled automatically when it is created.
location	location	<p>The location of the NVMe namespace within the ONTAP cluster. Valid in POST.</p> <p>NVMe namespaces do not support rename, or movement between volumes.</p> <ul style="list-style-type: none"> • readCreate: 1
name	string	<p>The fully qualified path name of the NVMe namespace composed of a "/vol" prefix, the volume name, the (optional) qtree name and base name of the namespace. Valid in POST.</p> <p>NVMe namespaces do not support rename, or movement between volumes.</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
space	space	The storage space related properties of the NVMe namespace.
status	status	Status information about the NVMe namespace.
subsystem_map	subsystem_map	<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 cost to retrieving property values for <code>subsystem_map</code>. They are not populated for either a collection GET or an instance GET unless explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>
svm	svm	
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",
  "create_time": "2018-06-04 19:00:00 UTC",
  "location": {
    "namespace": "namespace1",
    "qtree": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "id": 1,
      "name": "qt1"
    },
    "volume": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "volume1",
      "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
    }
  },
  "name": "/vol/volume1/qtree1/namespace1",
  "os_type": "linux",
  "space": {
    "block_size": 512,
    "size": 1073741824,
    "used": 0
  },
  "status": {
    "container_state": "online",

```

```
    "state": "online"
  },
  "subsystem_map": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "anagrpId": "00103050h",
    "nsid": "00000001h",
    "subsystem": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "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

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
13565952	The namespace clone request failed.
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.

Name	Type	Description
error	error	

Example error

```

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

```

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

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	The fully qualified path name of the clone source NVMe namespace composed of a "/vol" prefix, the volume name, the (optional) qtree name and base name of the namespace. Valid in POST and PATCH.
uuid	string	The unique identifier of the clone source NVMe namespace. Valid in POST and PATCH.

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	source	<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>

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
_links	_links	
id	integer	The identifier for the qtree, unique within the qtree's volume.
name	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
_links	_links	
name	string	The name of the volume.

Name	Type	Description
uuid	string	<p>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.</p> <ul style="list-style-type: none"> • example: 028baa66-41bd-11e9-81d5-00a0986138f7

location

The location of the NVMe namespace within the ONTAP cluster. Valid in POST.

NVMe namespaces do not support rename, or movement between volumes.

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>
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>

Name	Type	Description
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>

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"> • readOnly: 1

Name	Type	Description
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.</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>
guarantee	guarantee	Properties that request and report the space guarantee for the NVMe namespace.
size	integer	<p>The total provisioned size of the NVMe namespace.</p> <p>NVMe namespaces do not support resize.</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"> • example: 1073741824 • readCreate: 1

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"> • readOnly: 1

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.

Name	Type	Description
mapped	boolean	Reports if the NVMe namespace is mapped to an NVMe subsystem. There is an added cost to retrieving this property's value. It is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields 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.

subsystem

The NVMe subsystem to which the NVMe namespace is mapped.

Name	Type	Description
_links	_links	
name	string	The name of the NVMe subsystem.
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 cost to retrieving property values for `subsystem_map`. They are not populated for either a collection GET or an instance GET unless explicitly requested using the `fields` query parameter. See [DOC Requesting specific fields](#) to learn more.

Name	Type	Description
_links	_links	

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

svm

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

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.

In ONTAP, an NVMe namespace is located within a volume. Optionally, it can be located within a qtree in a volume.

An NVMe namespace is created to a specified size using thin or thick provisioning as determined by the volume on which it is created. NVMe namespaces support being cloned. An NVMe namespace cannot be renamed, resized, 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. See the NVMe namespace object model to learn more about each of the properties supported by the NVMe namespace REST API.

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.

Name	Type	Description
_links	_links	
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 cost to retrieving this property's value. It is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>

Name	Type	Description
clone	clone	<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.
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, you can check the <code>state</code> property to determine what error disabled the namespace. An NVMe namespace is enabled automatically when it is created.
location	location	<p>The location of the NVMe namespace within the ONTAP cluster. Valid in POST.</p> <p>NVMe namespaces do not support rename, or movement between volumes.</p> <ul style="list-style-type: none"> • <code>readCreate: 1</code>

Name	Type	Description
name	string	<p>The fully qualified path name of the NVMe namespace composed of a "/vol" prefix, the volume name, the (optional) qtree name and base name of the namespace. Valid in POST.</p> <p>NVMe namespaces do not support rename, or movement between volumes.</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>
space	space	The storage space related properties of the NVMe namespace.
status	status	Status information about the NVMe namespace.
subsystem_map	subsystem_map	<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 cost to retrieving property values for <code>subsystem_map</code>. They are not populated for either a collection GET or an instance GET unless explicitly requested using the <code>fields</code> query parameter. See DOC Requesting specific fields to learn more.</p>
svm	svm	
uuid	string	The unique identifier of the NVMe namespace.

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Copyright information

Copyright © 2024 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.