



Networking

REST API reference

NetApp
September 12, 2025

This PDF was generated from https://docs.netapp.com/us-en/ontap-restapi-98/networking_overview.html on September 12, 2025. Always check docs.netapp.com for the latest.

Table of Contents

| | |
|--|-----|
| Networking | 1 |
| Networking overview | 1 |
| Overview | 1 |
| IPspaces | 1 |
| Ethernet | 1 |
| Fibre Channel | 2 |
| IP | 2 |
| Manage broadcast domains | 3 |
| Network Ethernet broadcast-domains endpoint overview | 3 |
| Retrieve broadcast domains for the entire cluster | 13 |
| Create a new broadcast domain | 19 |
| Delete a broadcast domain | 24 |
| Retrieve broadcast domain details | 26 |
| Update broadcast domain properties | 31 |
| Manage network Ethernet ports | 35 |
| Network Ethernet ports endpoint overview | 35 |
| Retrieve ports | 55 |
| Create a new VLAN or LAG | 73 |
| Delete a VLAN or LAG | 88 |
| Retrieve a physical port, VLAN, or LAG details | 90 |
| Update a port | 103 |
| Retrieve historical port performance metrics | 116 |
| Retrieve network Ethernet switch ports | 124 |
| Network Ethernet switch ports endpoint overview | 124 |
| Retrieve Ethernet switch ports | 130 |
| Retrieve an Ethernet switch port | 140 |
| Manage Ethernet switches | 148 |
| Network Ethernet switches endpoint overview | 148 |
| Retrieve Ethernet switches attached to a chassis | 153 |
| Retrieve Ethernet switch details | 159 |
| Update an Ethernet switch | 164 |
| Manage FC network interfaces | 169 |
| Network FC interfaces endpoint overview | 169 |
| Retrieve FC interfaces | 183 |
| Create an FC interface | 205 |
| Delete an FC interface | 222 |
| Retrieve an FC interface | 224 |
| Update an FC interface | 239 |
| Retrieve FC interface historical performance metrics | 254 |
| Retrieve FC port information | 262 |
| Network FC ports endpoint overview | 262 |
| Retrieve FC ports | 270 |
| Retrieve an FC port | 290 |

| | |
|---|-----|
| Update an FC port | 303 |
| Retrieve FC port historical performance metrics | 317 |
| Manage HTTP proxy configuration | 326 |
| Network http-proxy endpoint overview | 326 |
| Retrieve HTTP proxy configurations for all SVMs and cluster IPspaces | 329 |
| Create an HTTP proxy configuration for an SVM or cluster IPspace | 335 |
| Deletes the http proxy configuration of the specified svm or cluster ipspace | 340 |
| Related ONTAP commands | 340 |
| Parameters | 340 |
| Response | 341 |
| Error | 341 |
| Definitions | 341 |
| Displays the http proxy server, port, and ipspace of the specified svm or cluster ipspace | 342 |
| Related ONTAP commands | 342 |
| Parameters | 342 |
| Response | 343 |
| Error | 344 |
| Definitions | 345 |
| Updates the proxy server, port, username, and password parameters | 347 |
| Related ONTAP commands | 347 |
| Parameters | 347 |
| Request Body | 347 |
| Response | 348 |
| Error | 348 |
| Definitions | 349 |
| Manage BGP peer groups | 351 |
| Network IP BGP peer-groups endpoint overview | 351 |
| Retrieve all BGP peer group details for VIP | 359 |
| Create a new BGP peer group for VIP | 367 |
| Delete a BGP peer group for VIP | 374 |
| Retrieve details of a BGP peer group for VIP | 376 |
| Update a BGP peer group for VIP | 382 |
| Manage network IP interfaces | 387 |
| Network IP interfaces endpoint overview | 387 |
| Retrieve all IP interface details | 403 |
| Create a new cluster-scoped or SVM-scoped interface | 419 |
| Delete an IP interface | 432 |
| Retrieve details for an IP interface | 433 |
| Update an IP interface | 444 |
| Retrieve interface historical performance metrics | 455 |
| Manage network IP routes | 463 |
| Network IP routes endpoint overview | 463 |
| Retrieve IP routes | 470 |
| Create a cluster-scoped or SVM-scoped static route | 477 |
| Delete an IP route | 482 |

| | |
|--|-----|
| Retrieve details for an IP route | 484 |
| Manage network IP service policies | 489 |
| Network IP service-policies endpoint overview | 489 |
| Retrieve service policies | 502 |
| Create a service policy for network interfaces | 508 |
| Delete a service policy for network interfaces | 513 |
| Retrieve a service policy | 515 |
| Update a service policy for network interfaces | 520 |
| Manage network IPspaces | 523 |
| Network ipspaces endpoint overview | 523 |
| Retrieve IPspaces for a cluster | 527 |
| Create a new domain with unique IP addresses | 532 |
| Delete an IPspace object | 534 |
| Retrieve information about an IPspace | 535 |
| Update an IPspace object | 537 |

Networking

Networking overview

Overview

The ONTAP networking APIs enable reporting on networking information, such as IPspaces, interfaces, routes, ports, service policies and broadcast domains. Some can also be used to manage networking.

IPspaces

IPspaces enable you to configure a single ONTAP cluster so that it can be accessed by clients from more than one administratively separate network domain, even if those clients are using the same IP address subnet range. This allows for separation of client traffic for privacy and security.

An IPspace defines a distinct IP address space in which storage virtual machines (SVMs) reside. Ports and IP addresses defined for an IPspace are applicable only within that IPspace. A distinct routing table is maintained for each SVM within an IPspace, so that no cross-SVM or cross-IPspace traffic routing occurs.

Ethernet

Broadcast Domains

A broadcast domain is a set of ports which would all receive a broadcast packet that is sent from any of the ports.

By accurately representing the physical network with an ONTAP broadcast domain, ONTAP ensures that IP interfaces are able to migrate to appropriate ports in case of failure. ONTAP also ensures that characteristics, such as MTU, stay matched across all ports of the broadcast domain.

A broadcast domain resides in an IPspace, and can be used by cluster-scoped or SVM-scoped IP interfaces in that IPspace. The scope of the broadcast domain's uniqueness is the IPspace it is in. You must create as many broadcast domains in an IPspace as there are IP subnets with interfaces in that IPspace.

Ports are mapped to an IPspace by assigning the port's broadcast domain.

Ports

A port is a physical or virtual Ethernet network device. Physical ports may be combined into Link Aggregation Groups (LAGs, or ifgrps), or divided into Virtual LANs (VLANs).

The GET and PATCH APIs are available for all port types. The POST and DELETE APIs are available for "lag" and "vlan" port types.

A given port can host zero or more IP interfaces.

A port exists in a broadcast domain and all ports within the same broadcast domain must have layer 2 network connectivity to one another. If a port within a broadcast domain goes down, any IP interfaces hosted by that port can fail over to other ports in the same broadcast domain.

Fibre Channel

Interfaces

Fibre Channel (FC) interfaces are the logical endpoints for Fibre Channel network connections to an SVM. A Fibre Channel interface provides Fibre Channel access to storage within the interface's SVM using either Fibre Channel Protocol (FCP) or Non-Volatile Memory Express over Fibre Channel (NVMe over FC).

The Fibre Channel interface REST API allows you to create, delete, update and discover Fibre Channel interfaces and obtain status information for Fibre Channel interfaces.

A Fibre Channel interface is created on a Fibre Channel port that is located on a cluster node. The Fibre Channel port must be specified to identify the location of the interface for a POST or PATCH that relocates an interface. You can identify the port by either supplying the node and port names or the port UUID.

Ports

Fibre Channel ports are the physical ports of Fibre Channel adapters on ONTAP cluster nodes that can be connected to Fibre Channel networks to provide Fibre Channel network connectivity. A Fibre Channel port defines the location of a Fibre Channel interface within the ONTAP cluster.

The Fibre Channel port REST API allows you to discover Fibre Channel ports, obtain status information for Fibre Channel ports, and configure Fibre Channel port properties.

Learn More

- *Fibre Channel Logins* found in both the SAN and NVMe sections. Fibre Channel logins represent connections formed by Fibre Channel initiators that have successfully logged in to ONTAP.

IP

Interfaces

An interface (also referred to as a *LIF* in ONTAP documentation) represents a network access point to a node in a cluster. In other words, an interface is essentially an IPv4 or IPv6 address with associated attributes.

IP interfaces are configured on ports to send and receive communications over the network. The port that will host the interface can either be explicitly specified using node and/or port fields or implicitly specified using a broadcast domain.

The IPspace of the interface is required for cluster-scoped interfaces. The SVM owning the interface is required for SVM-scoped interfaces. These interfaces are in the SVM's IPspace.

The service policy of an interface defines what network services are provided by the interface.

Routes

Routes indicate which IPv4 or IPv6 gateway to use to communicate with hosts that are not on the local subnet. Typically, an IP interface (or LIF) can only use a gateway if it has the same address family and is in the LIF's subnet.

It is important that every gateway address belongs to a physical or virtual router that has connectivity to the specified destination network.

SVM-scoped routes can only be used by IP interfaces of the specified SVM. Likewise, cluster-scoped routes can only be used by cluster-scoped IP interfaces in the specified IPspace.

Service Policies

Service policies are named groupings that define what services are supported by an IP interface. These include both built-in service policies (for example: default-data-files or default-management) and custom service policies.

Service policies are scoped to either an SVM or IPspace.

Manage broadcast domains

Network Ethernet broadcast-domains endpoint overview

Overview

A broadcast domain is a collection of Ethernet ports that have layer 2 connectivity. They are used to determine which Ethernet ports can host interfaces of various types. The broadcast domain REST API allows you to retrieve, create, modify, and delete broadcast domains. The broadcast domain APIs do not manage port membership. To add a port to a broadcast domain or to move a port to a different broadcast domain, use PATCH /network/ethernet/ports/<uuid>.</uuid>

Retrieving network Ethernet broadcast domain information

The broadcast domains GET API retrieves and displays relevant information pertaining to the broadcast domains configured in the cluster. The API retrieves the list of all broadcast domains configured in the cluster, or a specific broadcast domain.

Examples

Retrieving all broadcast domains in the cluster

The following output shows the list of all broadcast domains configured in a cluster.

```
# The API:  
/api/network/ethernet/broadcast-domains  
  
# The call:  
curl -X GET "https://<mgmt-ip>/api/network/ethernet/broadcast-domains" -H  
"accept: application/hal+json"  
  
# The response:  
{  
"records": [  
]
```

```
{
  "uuid": "6970c2a9-f34f-11e8-8373-005056bb6b85",
  "name": "Cluster",
  "ipspace": {
    "uuid": "6267eff8-f34f-11e8-8373-005056bb6b85",
    "name": "Cluster",
    "_links": {
      "self": {
        "href": "/api/network/ipspaces/6267eff8-f34f-11e8-8373-005056bb6b85"
      }
    }
  },
  "ports": [
    {
      "uuid": "626b4d19-f34f-11e8-8373-005056bb6b85",
      "name": "e0a",
      "node": {
        "name": "examplecluster-node01"
      },
      "_links": {
        "self": {
          "href": "/api/network/ethernet/ports/626b4d19-f34f-11e8-8373-005056bb6b85"
        }
      }
    },
    {
      "uuid": "626b77b9-f34f-11e8-8373-005056bb6b85",
      "name": "e0b",
      "node": {
        "name": "examplecluster-node01"
      },
      "_links": {
        "self": {
          "href": "/api/network/ethernet/ports/626b77b9-f34f-11e8-8373-005056bb6b85"
        }
      }
    }
  ],
  "mtu": 9000,
  "_links": {
    "self": {
      "href": "/api/network/ethernet/broadcast-domains/6970c2a9-f34f-11e8-8373-005056bb6b85"
    }
  }
}
```

```

        }
    }
},
{
    "uuid": "6972416c-f34f-11e8-8373-005056bb6b85",
    "name": "Default",
    "ipspace": {
        "uuid": "5f650349-f34f-11e8-8373-005056bb6b85",
        "name": "Default",
        "_links": {
            "self": {
                "href": "/api/network/ipspaces/5f650349-f34f-11e8-8373-005056bb6b85"
            }
        }
    },
    "ports": [
        {
            "uuid": "626bae19-f34f-11e8-8373-005056bb6b85",
            "name": "e0c",
            "node": {
                "name": "examplecluster-node01"
            },
            "_links": {
                "self": {
                    "href": "/api/network/ethernet/ports/626bae19-f34f-11e8-8373-005056bb6b85"
                }
            }
        },
        {
            "uuid": "626bd677-f34f-11e8-8373-005056bb6b85",
            "name": "e0d",
            "node": {
                "name": "examplecluster-node01"
            },
            "_links": {
                "self": {
                    "href": "/api/network/ethernet/ports/626bd677-f34f-11e8-8373-005056bb6b85"
                }
            }
        }
    ],
    "mtu": 1500,
    "_links": {

```

```

    "self": {
      "href": "/api/network/ethernet/broadcast-domains/6972416c-f34f-
11e8-8373-005056bb6b85"
    }
  }
],
"num_records": 2,
"_links": {
  "self": {
    "href": "/api/network/ethernet/broadcast-domains?fields=*&"
  }
}
}

```

Retrieving a specific broadcast domain

The following output shows the response returned when a specific broadcast domain is requested. The system returns an error if there is no broadcast domain with the requested UUID.

```

# The API:
/api/network/ethernet/broadcast-domains/{uuid}

# The call:
curl -X GET "https://<mgmt-ip>/api/network/ethernet/broadcast-
domains/4475a2c8-f8a0-11e8-8d33-005056bb986f/?fields=*&" -H "accept:
application/hal+json"

# The response:
{
  "uuid": "4475a2c8-f8a0-11e8-8d33-005056bb986f",
  "name": "Cluster",
  "ipspace": {
    "uuid": "3e518ed5-f8a0-11e8-8d33-005056bb986f",
    "name": "Cluster",
    "_links": {
      "self": {
        "href": "/api/network/ipspaces/3e518ed5-f8a0-11e8-8d33-005056bb986f"
      }
    }
  },
  "ports": [
    {

```

```

"uuid": "3e539a62-f8a0-11e8-8d33-005056bb986f",
"name": "e0a",
"node": {
  "name": "examplecluster-node01"
},
"_links": {
  "self": {
    "href": "/api/network/ethernet/ports/3e539a62-f8a0-11e8-8d33-005056bb986f"
  }
},
{
  "uuid": "3e53c94a-f8a0-11e8-8d33-005056bb986f",
  "name": "e0b",
  "node": {
    "name": "examplecluster-node01"
  },
  "_links": {
    "self": {
      "href": "/api/network/ethernet/ports/3e53c94a-f8a0-11e8-8d33-005056bb986f"
    }
  }
},
],
"mtu": 9000,
"_links": {
  "self": {
    "href": "/api/network/ethernet/broadcast-domains/4475a2c8-f8a0-11e8-8d33-005056bb986f/"
  }
}
}

```

Retrieving all broadcast domains with a specific name

The following output shows the response returned when broadcast domains with a specific name in any IPspace are requested.

```

# The API:
/api/network/ethernet/broadcast-domains

# The call:
curl -X GET "https://10.224.87.121/api/network/ethernet/broadcast-
domains/?name=bd1" -H "accept: application/hal+json"

# The response:
{
  "records": [
    {
      "uuid": "66b607e5-4bee-11e9-af6a-005056bb13c0",
      "name": "bd1",
      "_links": {
        "self": {
          "href": "/api/network/ethernet/broadcast-domains/66b607e5-4bee-
11e9-af6a-005056bb13c0"
        }
      }
    }
  ],
  "num_records": 1,
  "_links": {
    "self": {
      "href": "/api/network/ethernet/broadcast-domains/?name=bd1"
    }
  }
}

```

Retrieving the broadcast domains for an IPspace

The following output shows the response returned when the broadcast domains for a specified IPspace are requested.

```

# The API:
/api/network/ethernet/broadcast-domains

# The call:
curl -X GET "https://10.224.87.121/api/network/ethernet/broadcast-
domains/?ipspace.name=Cluster&fields=*" -H "accept: application/hal+json"

# The response:

```

```
{
  "records": [
    {
      "uuid": "ae69070c-4bed-11e9-af6a-005056bb13c0",
      "name": "Cluster",
      "ipspace": {
        "uuid": "ac466a88-4bed-11e9-af6a-005056bb13c0",
        "name": "Cluster",
        "_links": {
          "self": {
            "href": "/api/network/ipspaces/ac466a88-4bed-11e9-af6a-005056bb13c0"
          }
        }
      },
      "ports": [
        {
          "uuid": "acd67884-4bed-11e9-af6a-005056bb13c0",
          "name": "e0a",
          "node": {
            "name": "examplecluster-node-1"
          },
          "_links": {
            "self": {
              "href": "/api/network/ethernet/ports/acd67884-4bed-11e9-af6a-005056bb13c0"
            }
          }
        },
        {
          "uuid": "ace1a36f-4bed-11e9-af6a-005056bb13c0",
          "name": "e0b",
          "node": {
            "name": "examplecluster-node-1"
          },
          "_links": {
            "self": {
              "href": "/api/network/ethernet/ports/ace1a36f-4bed-11e9-af6a-005056bb13c0"
            }
          }
        }
      ],
      "mtu": 1500,
      "_links": {
        "self": {

```

```
        "href": "/api/network/ethernet/broadcast-domains/ae69070c-4bed-  
11e9-af6a-005056bb13c0"  
    }  
}  
}  
]  
,"num_records": 1,  
"_links": {  
    "self": {  
        "href": "/api/network/ethernet/broadcast-  
domains/?ipspace.name=Cluster&fields=*"  
    }  
}  
}
```

Creating network Ethernet broadcast domains

You can use the POST API to create broadcast domains.

Example

Creating a new broadcast domain

The following example shows how to create a broadcast domain with a name of 'bd1' and an MTU of 1500.

```

# The API:
/api/network/ethernet/broadcast-domains

# The call:
curl -X POST "https://<mgmt-ip>/api/network/ethernet/broadcast-
domains?return_records=true" -H "accept: application/hal+json" -d '{
"name": "bd1", "mtu": 1500 }'

# The response:
{
  "num_records": 1,
  "records": [
    {
      "name": "bd1",
      "mtu": 1500,
      "_links": {
        "self": {
          "href": "/api/network/ethernet/broadcast-domains/"
        }
      }
    }
  ]
}

```

Updating network Ethernet broadcast domains

You can use the PATCH API to update the attributes of broadcast domains.

Examples

Updating the name and MTU of a specific broadcast domain

The following example shows how the PATCH request changes the broadcast domain name to 'bd2' and the broadcast domain MTU to 9000.

```
# The API:  
/api/network/ethernet/broadcast-domains/{uuid}  
  
# The call:  
curl -X PATCH "https://<mgmt-ip>/api/network/ethernet/broadcast-  
domains/6cde03b2-f8a2-11e8-8d33-005056bb986f/" -d '{ "name": "bd2", "mtu":  
9000 }'  
{  
}
```

Updating the IPspace of a specific broadcast domain

The following example shows how the PATCH request changes the IPspace of a broadcast domain to 'ipspace2'.

```
# The API:  
/api/network/ethernet/broadcast-domains/{uuid}  
  
# The call:  
curl -X PATCH "https://<mgmt-ip>/api/network/ethernet/broadcast-  
domains/c6fe2541-61f4-11e9-a66e-005056bbe83e" -d '{ "ipspace" : { "name" :  
"ipspace2" } }'  
{  
}
```

Deleting network Ethernet broadcast domains

You can use the DELETE API to delete a broadcast domain from the cluster configuration.

Example

Deleting a specific broadcast domain

The following DELETE request deletes a broadcast domain.

```
# The API:  
/api/network/ethernet/broadcast-domains/{uuid}  
  
# The call:  
curl -X DELETE "https://<mgmt-ip>/api/network/ethernet/broadcast-  
domains/6cde03b2-f8a2-11e8-8d33-005056bb986f/"
```

Retrieve broadcast domains for the entire cluster

GET /network/ethernet/broadcast-domains

Introduced In: 9.6

Retrieves a collection of broadcast domains for the entire cluster.

Related ONTAP commands

- `network port broadcast-domain show`

Parameters

| Name | Type | In | Required | Description |
|-----------------|---------------|-------|----------|-------------------------------|
| ports.name | string | query | False | Filter by ports.name |
| ports.node.name | string | query | False | Filter by ports.node.name |
| ports.uuid | string | query | False | Filter by ports.uuid |
| name | string | query | False | Filter by name |
| ipspace.name | string | query | False | Filter by ipspace.name |
| ipspace.uuid | string | query | False | Filter by ipspace.uuid |
| uuid | string | query | False | Filter by uuid |
| mtu | integer | query | False | Filter by mtu |
| fields | array[string] | query | False | Specify the fields to return. |

| Name | Type | In | Required | Description |
|----------------|---------------|-------|----------|---|
| max_records | integer | query | False | Limit the number of records returned. |
| return_records | boolean | query | False | The default is true for GET calls. When set to false, only the number of records is returned. <ul style="list-style-type: none"> Default value: 1 |
| return_timeout | integer | query | False | The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached. <ul style="list-style-type: none"> Default value: 1 Max value: 120 Min value: 0 |
| order_by | array[string] | query | False | Order results by specified fields and optional [asc] |

Response

Status: 200, Ok

| Name | Type | Description |
|-------------|-------------------------|-------------|
| _links | _links | |
| num_records | integer | |
| records | array[broadcast_domain] | |

Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "ipsspace": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        }
      },
      "name": "exchange",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "mtu": "1500",
    "name": "bd1",
    "ports": [
      {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "e1b",
        "node": {
          "name": "node1"
        },
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      }
    ],
    "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 | |

ipspace

Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input.

| Name | Type | Description |
|--------|------------------------|--------------|
| _links | _links | |
| name | string | IPspace name |
| uuid | string | IPspace UUID |

node

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

ports

Port UUID along with readable names

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| node | node | |
| uuid | string | |

broadcast_domain

Set of ports that will receive a broadcast Ethernet packet from any of them

| Name | Type | Description |
|---------|--------------------------------|---|
| _links | _links | |
| ipspace | ipspace | Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input. |
| mtu | integer | Maximum transmission unit, largest packet size on this network |
| name | string | Name of the broadcast domain, scoped to its IPspace |
| ports | array[ports] | Ports that belong to the broadcast domain |
| uuid | string | Broadcast domain UUID |

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 a new broadcast domain

POST /network/ethernet/broadcast-domains

Introduced In: 9.6

Creates a new broadcast domain.

Required properties

- `name` - Name of the broadcast-domain to create.
- `mtu` - Maximum transmission unit (MTU) of the broadcast domain.

Recommended optional properties

- `ipspace.name` or `ipspace.uuid` - IPspace the broadcast domain belongs to.

Default property values

If not specified in POST, the following default property values are assigned:

- `ipspace` - *Default*

Related ONTAP commands

- `network port broadcast-domain create`

Parameters

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

Request Body

| Name | Type | Description |
|---------|--------------------------------|---|
| ipspace | ipspace | Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input. |
| mtu | integer | Maximum transmission unit, largest packet size on this network |
| name | string | Name of the broadcast domain, scoped to its IPspace |
| ports | array[ports] | Ports that belong to the broadcast domain |

| Name | Type | Description |
|------|--------|-----------------------|
| uuid | string | Broadcast domain UUID |

Example request

```
{
  "ipspace": {
    "name": "exchange",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "mtu": "1500",
  "name": "bd1",
  "ports": [
    {
      "name": "e1b",
      "node": {
        "name": "node1"
      },
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
  ],
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
```

Response

Status: 201, Created

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|---|
| 1377267 | The specified IPspace does not exist. |
| 1967082 | The specified ipspace.name does not match the IPspace name of ipspace.uuid. |
| 1967102 | A POST operation might have left the configuration in an inconsistent state. Check the configuration. |

| Error Code | Description |
|-------------------|--|
| 53281982 | The specified broadcast domain name is reserved by the system. |

Definitions

See Definitions

href

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

_links

ipspace

Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input.

| Name | Type | Description |
|------|--------|--------------|
| name | string | IPspace name |
| uuid | string | IPspace UUID |

node

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

ports

Port UUID along with readable names

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| node | node | |
| uuid | string | |

broadcast_domain

Set of ports that will receive a broadcast Ethernet packet from any of them

| Name | Type | Description |
|---------|---------|---|
| ipspace | ipspace | Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input. |
| mtu | integer | Maximum transmission unit, largest packet size on this network |

| Name | Type | Description |
|-------|--------------------------------|---|
| name | string | Name of the broadcast domain, scoped to its IPspace |
| ports | array[ports] | Ports that belong to the broadcast domain |
| uuid | string | Broadcast domain UUID |

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 a broadcast domain

`DELETE /network/ethernet/broadcast-domains/{uuid}`

Introduced In: 9.6

Deletes a broadcast domain.

Related ONTAP commands

- `network port broadcast-domain delete`

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|-----------------------|
| uuid | string | path | True | Broadcast domain UUID |

Response

Status: 200, Ok

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|--|
| 1967103 | A broadcast domain with ports cannot 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 broadcast domain details

GET /network/ethernet/broadcast-domains/{uuid}

Introduced In: 9.6

Retrieves details of a broadcast domain.

Related ONTAP commands

- `network port broadcast-domain show`

Parameters

| Name | Type | In | Required | Description |
|--------|---------------|-------|----------|-------------------------------|
| uuid | string | path | True | Broadcast domain UUID |
| fields | array[string] | query | False | Specify the fields to return. |

Response

Status: 200, Ok

| Name | Type | Description |
|---------|--------------------------------|---|
| _links | _links | |
| ipspace | ipspace | Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input. |
| mtu | integer | Maximum transmission unit, largest packet size on this network |
| name | string | Name of the broadcast domain, scoped to its IPspace |
| ports | array[ports] | Ports that belong to the broadcast domain |
| uuid | string | Broadcast domain UUID |

Example response

```
{  
  "_links": {  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "ipspace": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    },  
    "name": "exchange",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "mtu": "1500",  
  "name": "bd1",  
  "ports": [  
    {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "name": "elb",  
      "node": {  
        "name": "node1"  
      },  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    }  
  ],  
  "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 | |

ipspace

Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input.

| Name | Type | Description |
|--------|--------|--------------|
| _links | _links | |
| name | string | IPspace name |
| uuid | string | IPspace UUID |

node

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

ports

Port UUID along with readable names

| Name | Type | Description |
|--------|--------|-------------|
| _links | _links | |
| name | string | |
| node | node | |
| uuid | string | |

error_arguments

| Name | Type | Description |
|------|--------|---------------|
| code | string | Argument code |

| Name | Type | Description |
|-----------|------------------------|---|
| 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 broadcast domain properties

PATCH /network/ethernet/broadcast-domains/{uuid}

Introduced In: 9.6

Updates the properties of a broadcast domain.

Related ONTAP commands

- `network port broadcast-domain modify`
- `network port broadcast-domain rename`
- `network port broadcast-domain move`

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|-----------------------|
| uuid | string | path | True | Broadcast domain UUID |

Request Body

| Name | Type | Description |
|---------|---------|---|
| ipspace | ipspace | Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input. |

| Name | Type | Description |
|-------|--------------|--|
| mtu | integer | Maximum transmission unit, largest packet size on this network |
| name | string | Name of the broadcast domain, scoped to its IPspace |
| ports | array[ports] | Ports that belong to the broadcast domain |
| uuid | string | Broadcast domain UUID |

Example request

```
{
  "ipspace": {
    "name": "exchange",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "mtu": "1500",
  "name": "bd1",
  "ports": [
    {
      "name": "elb",
      "node": {
        "name": "node1"
      },
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
  ],
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
```

Response

```
Status: 200, Ok
```

Error

```
Status: Default
```

ONTAP Error Response Codes

| Error Code | Description |
|------------|--|
| 1377267 | The specified IPspace does not exist. |
| 1377269 | Failed to lookup the specified IPspace. |
| 1377560 | Broadcast domain already exists in specified IPspace. |
| 1377605 | Moving the system-generated broadcast domain to another IPspace is not supported. |
| 1967082 | The specified ipspace.name does not match the IPspace name of ipspace.uuid. |
| 1967150 | The specified ipspace.uuid is not valid. |
| 1967151 | The specified ipspace.uuid and ipspace.name do not match. |
| 1967152 | Patching IPspace for a broadcast domain requires an effective cluster version of 9.7 or later. |

Definitions

See Definitions

href

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

_links

ipspace

Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input.

| Name | Type | Description |
|------|--------|--------------|
| name | string | IPspace name |
| uuid | string | IPspace UUID |

node

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

ports

Port UUID along with readable names

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| node | node | |
| uuid | string | |

broadcast_domain

Set of ports that will receive a broadcast Ethernet packet from any of them

| Name | Type | Description |
|---------|---------|---|
| ipspace | ipspace | Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input. |
| mtu | integer | Maximum transmission unit, largest packet size on this network |

| Name | Type | Description |
|-------|--------------------------------|---|
| name | string | Name of the broadcast domain, scoped to its IPspace |
| ports | array[ports] | Ports that belong to the broadcast domain |
| uuid | string | Broadcast domain UUID |

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 network Ethernet ports

Network Ethernet ports endpoint overview

Overview

A port is a physical or virtual Ethernet network device. Physical ports may be combined into Link Aggregation Groups (LAGs or ifgrps), or divided into Virtual LANs (VLANs).

GET (collection), GET (instance), and PATCH APIs are available for all port types. POST and DELETE APIs are available for "lag" (ifgrp) and "vlan" port types.

Retrieving network port information

The network ports GET API retrieves and displays relevant information pertaining to the ports configured in the cluster. The API retrieves the list of all ports configured in the cluster, or specifically requested ports. The fields

returned in the response vary for different ports and configurations.

Examples

Retrieving all ports in the cluster

The following output displays the UUID, name, and port type for all ports configured in a 2-node cluster. The port types are physical, vlan, and lag (ifgrp).

```
# The API:  
/api/network/ethernet/ports  
  
# The call:  
curl -X GET "https://<mgmt-  
ip>/api/network/ethernet/ports?fields=uuid,name,type" -H "accept:  
application/hal+json"  
  
# The response:  
{  
  "records": [  
    {  
      "uuid": "2d2c90c0-f70d-11e8-b145-005056bb5b8e",  
      "name": "e0a",  
      "type": "physical",  
      "_links": {  
        "self": {  
          "href": "/api/network/ethernet/ports/2d2c90c0-f70d-11e8-b145-  
005056bb5b8e"  
        }  
      }  
    },  
    {  
      "uuid": "2d3004da-f70d-11e8-b145-005056bb5b8e",  
      "name": "e0b",  
      "type": "physical",  
      "_links": {  
        "self": {  
          "href": "/api/network/ethernet/ports/2d3004da-f70d-11e8-b145-  
005056bb5b8e"  
        }  
      }  
    },  
    {  
      "uuid": "2d34a2cb-f70d-11e8-b145-005056bb5b8e",  
      "name": "e0c",  
      "type": "physical",  
      "_links": {  
        "self": {  
          "href": "/api/network/ethernet/ports/2d34a2cb-f70d-11e8-b145-  
005056bb5b8e"  
        }  
      }  
    }  
  ]  
}
```

```

    "type": "physical",
    "_links": {
        "self": {
            "href": "/api/network/ethernet/ports/2d34a2cb-f70d-11e8-b145-
005056bb5b8e"
        }
    }
},
{
    "uuid": "2d37189f-f70d-11e8-b145-005056bb5b8e",
    "name": "e0d",
    "type": "physical",
    "_links": {
        "self": {
            "href": "/api/network/ethernet/ports/2d37189f-f70d-11e8-b145-
005056bb5b8e"
        }
    }
},
{
    "uuid": "35de5d8b-f70d-11e8-abdf-005056bb7fc8",
    "name": "e0a",
    "type": "physical",
    "_links": {
        "self": {
            "href": "/api/network/ethernet/ports/35de5d8b-f70d-11e8-abdf-
005056bb7fc8"
        }
    }
},
{
    "uuid": "35de78cc-f70d-11e8-abdf-005056bb7fc8",
    "name": "e0b",
    "type": "physical",
    "_links": {
        "self": {
            "href": "/api/network/ethernet/ports/35de78cc-f70d-11e8-abdf-
005056bb7fc8"
        }
    }
},
{
    "uuid": "35dead3c-f70d-11e8-abdf-005056bb7fc8",
    "name": "e0c",
    "type": "physical",
    "_links": {

```

```

    "self": {
      "href": "/api/network/ethernet/ports/35dead3c-f70d-11e8-abdf-
005056bb7fc8"
    }
  }
},
{
  "uuid": "35deda90-f70d-11e8-abdf-005056bb7fc8",
  "name": "e0d",
  "type": "physical",
  "_links": {
    "self": {
      "href": "/api/network/ethernet/ports/35deda90-f70d-11e8-abdf-
005056bb7fc8"
    }
  }
},
{
  "uuid": "42e25145-f97d-11e8-ade9-005056bb7fc8",
  "name": "e0c-100",
  "type": "vlan",
  "_links": {
    "self": {
      "href": "/api/network/ethernet/ports/42e25145-f97d-11e8-ade9-
005056bb7fc8"
    }
  }
},
{
  "uuid": "569e0abd-f97d-11e8-ade9-005056bb7fc8",
  "name": "a0a",
  "type": "lag",
  "_links": {
    "self": {
      "href": "/api/network/ethernet/ports/569e0abd-f97d-11e8-ade9-
005056bb7fc8"
    }
  }
},
],
"num_records": 10,
"_links": {
  "self": {
    "href": "/api/network/ethernet/ports?fields=uuid,name,type"
  }
}

```

}

Retrieving a specific physical port

The following output displays the response when a specific physical port is requested. The system returns an error when there is no port with the requested UUID. Also, the speed field is set only if the state of the port is up.

```

"enabled": true,
"state": "up",
"mtu": 1500,
"speed": "1000",
"reachability": "not_repairable",
"reachable_broadcast_domains": [
  {
    "uuid": "36434bec-f70d-11e8-b145-005056bb5b8e",
    "name": "Default",
    "ipspace": {
      "name": "Default"
    },
    "_links": {
      "self": {
        "href": "/api/network/ethernet/broadcast-domains/36434bec-f70d-11e8-b145-005056bb5b8e"
      }
    }
  },
  {
    "uuid": "df640ccf-72c4-11ea-b31d-005056bbfb29",
    "name": "Default-1",
    "ipspace": {
      "name": "Default"
    },
    "_links": {
      "self": {
        "href": "/api/network/ethernet/broadcast-domains/df640ccf-72c4-11ea-b31d-005056bbfb29"
      }
    }
  }
],
"_links": {
  "self": {
    "href": "/api/network/ethernet/ports/2d37189f-f70d-11e8-b145-005056bb5b8e"
  }
}
}

```

Retrieving a specific VLAN port

The following output displays the response when a specific VLAN port is requested. The system returns an error when there is no port with the requested UUID. Also, the speed field is set only if the state of the port is

up.

```
# The API:  
/api/network/ethernet/ports/{uuid}  
  
# The call:  
curl -X GET "https://<mgmt-ip>/api/network/ethernet/ports/42e25145-f97d-  
11e8-ade9-005056bb7fc8?fields=*" -H "accept: application/hal+json"  
  
# The response:  
{  
  "uuid": "42e25145-f97d-11e8-ade9-005056bb7fc8",  
  "name": "e0e-100",  
  "mac_address": "00:50:56:bb:52:2f",  
  "type": "vlan",  
  "node": {  
    "uuid": "6042cf47-f70c-11e8-abdf-005056bb7fc8",  
    "name": "user-cluster-02",  
    "_links": {  
      "self": {  
        "href": "/api/cluster/nodes/6042cf47-f70c-11e8-abdf-005056bb7fc8"  
      }  
    }  
  },  
  "broadcast_domain": {  
    "uuid": "36434bec-f70d-11e8-b145-005056bb5b8e",  
    "name": "Default",  
    "ipspace": {  
      "name": "Default"  
    },  
    "_links": {  
      "self": {  
        "href": "/api/network/ethernet/broadcast-domains/36434bec-f70d-11e8-  
b145-005056bb5b8e"  
      }  
    }  
  },  
  "enabled": true,  
  "state": "up",  
  "mtu": 1500,  
  "speed": "1000",  
  "reachability": "ok",  
  "reachable_broadcast_domains": [  
  ]  
}
```

```

"uuid": "36434bec-f70d-11e8-b145-005056bb5b8e",
"name": "Default",
"ipspace": {
  "name": "Default"
},
"_links": {
  "self": {
    "href": "/api/network/ethernet/broadcast-domains/36434bec-f70d-11e8-b145-005056bb5b8e"
  }
}
],
"vlan": {
  "tag": 100,
  "base_port": {
    "uuid": "35deff03-f70d-11e8-abdf-005056bb7fc8",
    "name": "e0e",
    "node": {
      "name": "user-cluster-02"
    },
    "_links": {
      "self": {
        "href": "/api/network/ethernet/ports/35deff03-f70d-11e8-abdf-005056bb7fc8"
      }
    }
  }
},
"_links": {
  "self": {
    "href": "/api/network/ethernet/ports/42e25145-f97d-11e8-ade9-005056bb7fc8"
  }
}
}

```

Retrieving a specific LAG port

The following output displays the response when a specific LAG port is requested. The system returns an error when there is no port with the requested UUID. Also, the speed and lag.active_ports fields are set only if the state of the port is up.

```

# The API:
/api/network/ethernet/ports/{uuid}

# The call:
curl -X GET "https://<mgmt-ip>/api/network/ethernet/ports/569e0abd-f97d-
11e8-ade9-005056bb7fc8?fields=*" -H "accept: application/hal+json"

# The response:
{
  "uuid": "569e0abd-f97d-11e8-ade9-005056bb7fc8",
  "name": "a0a",
  "mac_address": "02:50:56:bb:7f:c8",
  "type": "lag",
  "node": {
    "uuid": "6042cf47-f70c-11e8-abdf-005056bb7fc8",
    "name": "user-cluster-02",
    "_links": {
      "self": {
        "href": "/api/cluster/nodes/6042cf47-f70c-11e8-abdf-005056bb7fc8"
      }
    }
  },
  "broadcast_domain": {
    "uuid": "36434bec-f70d-11e8-b145-005056bb5b8e",
    "name": "Default",
    "ipspace": {
      "name": "Default"
    },
    "_links": {
      "self": {
        "href": "/api/network/ethernet/broadcast-domains/36434bec-f70d-11e8-
b145-005056bb5b8e"
      }
    }
  },
  "enabled": true,
  "state": "up",
  "mtu": 1500,
  "speed": "1000",
  "reachability": "repairable",
  "reachable_broadcast_domains": [
    {
      "uuid": "c7934b4f-691f-11ea-87fd-005056bb1ad3",
      "name": "Default",
      "ipspace": {
        "name": "Default"
      }
    }
  ]
}

```

```

} ,
"_links": {
  "self": {
    "href": "/api/network/ethernet/broadcast-domains/c7934b4f-691f-
11ea-87fd-005056bb1ad3"
  }
}
],
"lag": {
  "mode": "singlemode",
  "distribution_policy": "mac",
  "member_ports": [
    {
      "uuid": "35df318d-f70d-11e8-abdf-005056bb7fc8",
      "name": "e0f",
      "node": {
        "name": "user-cluster-02"
      },
      "_links": {
        "self": {
          "href": "/api/network/ethernet/ports/35df318d-f70d-11e8-abdf-
005056bb7fc8"
        }
      }
    },
    {
      "uuid": "35df5bad-f70d-11e8-abdf-005056bb7fc8",
      "name": "e0g",
      "node": {
        "name": "user-cluster-02"
      },
      "_links": {
        "self": {
          "href": "/api/network/ethernet/ports/35df5bad-f70d-11e8-abdf-
005056bb7fc8"
        }
      }
    },
    {
      "uuid": "35df9926-f70d-11e8-abdf-005056bb7fc8",
      "name": "e0h",
      "node": {
        "name": "user-cluster-02"
      },
      "_links": {

```

```

        "self": {
            "href": "/api/network/ethernet/ports/35df9926-f70d-11e8-abdf-
005056bb7fc8"
        }
    }
}

],
"active_ports": [
{
    "uuid": "35df318d-f70d-11e8-abdf-005056bb7fc8",
    "name": "e0f",
    "_links": {
        "self": {
            "href": "/api/network/ethernet/ports/35df318d-f70d-11e8-abdf-
005056bb7fc8"
        }
    }
}
]
},
"_links": {
    "self": {
        "href": "/api/network/ethernet/ports/569e0abd-f97d-11e8-ade9-
005056bb7fc8"
    }
}
}
}

```

Retrieving all LAG (ifgrp) ports in the cluster

This command retrieves all LAG ports in the cluster (that is, all ports with type=LAG). The example shows how to filter a GET collection based on type.

```

# The API:
/api/network/ethernet/ports

# The call:
curl -X GET "https://<mgmt-
ip>/api/network/ethernet/ports?type=lag&node.name=user-cluster-
01&fields=name,enabled,speed,mtu" -H "accept: application/hal+json"

# The response:
{

```

```

"records": [
  {
    "uuid": "0c226db0-4b63-11e9-8113-005056bbe040",
    "name": "a0b",
    "type": "lag",
    "node": {
      "name": "user-cluster-01"
    },
    "enabled": true,
    "mtu": 1500,
    "speed": "1000",
    "_links": {
      "self": {
        "href": "/api/network/ethernet/ports/0c226db0-4b63-11e9-8113-005056bbe040"
      }
    }
  },
  {
    "uuid": "d3a84153-4b3f-11e9-a00d-005056bbe040",
    "name": "a0a",
    "type": "lag",
    "node": {
      "name": "user-cluster-01"
    },
    "enabled": true,
    "mtu": 1500,
    "speed": "1000",
    "_links": {
      "self": {
        "href": "/api/network/ethernet/ports/d3a84153-4b3f-11e9-a00d-005056bbe040"
      }
    }
  }
],
"num_records": 2,
"_links": {
  "self": {
    "href": "/api/network/ethernet/ports?fields=name,enabled,speed,mtu&type=lag&node.name=user-cluster-01"
  }
}
}

```

Creating VLAN and LAG ports

You can use the network ports POST API to create VLAN and LAG ports. If you supply the optional broadcast domain property, the specified broadcast domain will be assigned to the new port immediately. Otherwise, within a few minutes automatic probing will determine the correct broadcast domain and will assign it to the port. During that period of time, the port will not be capable of hosting interfaces.

Examples

Creating a VLAN port

The following output displays the record returned after the creation of a VLAN port on "e0e" and VLAN tag "100".

```
# The API:  
/api/network/ethernet/ports  
  
# The call:  
curl -X POST "https://<mgmt-  
ip>/api/network/ethernet/ports?return_records=true" -H "accept:  
application/hal+json" -H "Content-Type: application/json" -d "{ \"type\":  
\"vlan\", \"node\": { \"name\": \"user-cluster-01\" }, \"enabled\": true,  
\"vlan\": { \"tag\": 100, \"base_port\": { \"name\": \"e0e\", \"node\": {  
\"name\": \"user-cluster-01\" } } } }"  
  
# The response:  
{  
  "num_records": 1,  
  "records": [  
    {  
      "uuid": "88b2f682-fa42-11e8-a6d7-005056bb5b8e",  
      "type": "vlan",  
      "node": {  
        "uuid": "faa56898-f70c-11e8-b145-005056bb5b8e",  
        "name": "user-cluster-01",  
        "_links": {  
          "self": {  
            "href": "/api/cluster/nodes/faa56898-f70c-11e8-b145-  
005056bb5b8e"  
          }  
        }  
      },  
      "enabled": true,  
      "vlan": {  
        "tag": 100,  
        "base_port": {  
          "name": "e0e",  
          "node": {  
            "name": "user-cluster-01"  
          }  
        }  
      }  
    }  
  ]  
}
```

```

"base_port": {
    "uuid": "2d39df72-f70d-11e8-b145-005056bb5b8e",
    "name": "e0e",
    "node": {
        "name": "user-cluster-01"
    },
    "_links": {
        "self": {
            "href": "/api/network/ethernet/ports/2d39df72-f70d-11e8-b145-005056bb5b8e"
        }
    }
},
"_links": {
    "self": {
        "href": "/api/network/ethernet/ports/88b2f682-fa42-11e8-a6d7-005056bb5b8e"
    }
}
}
]
}

```

Creating a VLAN port in a specific broadcast domain

The following output displays the record returned after the creation of a VLAN port on "e0e" and VLAN tag "100". Also, the VLAN port is added to the "Default" broadcast domain in the "Default" IPspace.

```

# The API:
/api/network/ethernet/ports

# The call:
curl -X POST "https://<mgmt-ip>/api/network/ethernet/ports?return_records=true" -H "accept: application/hal+json" -H "Content-Type: application/json" -d "{ \"type\": \"vlan\", \"node\": { \"name\": \"user-cluster-01\" }, \"broadcast_domain\": { \"name\": \"Default\", \"ipspace\": { \"name\": \"Default\" } }, \"enabled\": true, \"vlan\": { \"tag\": 100, \"base_port\": { \"name\": \"e0e\", \"node\": { \"name\": \"user-cluster-01\" } } } }"

# The response:

```

```
{
  "num_records": 1,
  "records": [
    {
      "uuid": "88b2f682-fa42-11e8-a6d7-005056bb5b8e",
      "type": "vlan",
      "node": {
        "uuid": "faa56898-f70c-11e8-b145-005056bb5b8e",
        "name": "user-cluster-01",
        "_links": {
          "self": {
            "href": "/api/cluster/nodes/faa56898-f70c-11e8-b145-005056bb5b8e"
          }
        }
      },
      "broadcast_domain": {
        "uuid": "36434bec-f70d-11e8-b145-005056bb5b8e",
        "name": "Default",
        "ipspace": {
          "name": "Default"
        },
        "_links": {
          "self": {
            "href": "/api/network/ethernet/broadcast-domains/36434bec-f70d-11e8-b145-005056bb5b8e"
          }
        }
      },
      "enabled": true,
      "vlan": {
        "tag": 100,
        "base_port": {
          "uuid": "2d39df72-f70d-11e8-b145-005056bb5b8e",
          "name": "e0e",
          "node": {
            "name": "user-cluster-01"
          },
          "_links": {
            "self": {
              "href": "/api/network/ethernet/ports/2d39df72-f70d-11e8-b145-005056bb5b8e"
            }
          }
        }
      }
    }
  ]
}
```

```

    "_links": {
        "self": {
            "href": "/api/network/ethernet/ports/88b2f682-fa42-11e8-a6d7-
005056bb5b8e"
        }
    }
}
]
}

```

Creating a LAG (ifgrp) port

The following output displays the record returned after the creation of a LAG port with "e0f", "e0g" and "e0h" as member ports.

```

# The API:
/api/network/ethernet/ports

# The call:
curl -X POST "https://<mgmt-
ip>/api/network/ethernet/ports?return_records=true" -H "accept:
application/json" -H "Content-Type: application/json" -d "{  \"type\":
\"lag\",  \"node\": { \"name\": \"user-cluster-01\" },  \"enabled\": true,
\"lag\": { \"mode\": \"singlemode\", \"distribution_policy\": \"mac\",
\"member_ports\": [ { \"name\": \"e0f\", \"node\": { \"name\": \"user-
cluster-01\" } }, { \"name\": \"e0g\", \"node\": { \"name\": \"user-
cluster-01\" } }, { \"name\": \"e0h\", \"node\": { \"name\": \"user-
cluster-01\" } } ] } }"

# The response:
{
"num_records": 1,
"records": [
{
    "uuid": "1807772a-fa4d-11e8-a6d7-005056bb5b8e",
    "type": "lag",
    "node": {
        "uuid": "faa56898-f70c-11e8-b145-005056bb5b8e",
        "name": "user-cluster-01"
    },
    "enabled": true,
    "lag": {
        "mode": "singlemode",

```

```

  "distribution_policy": "mac",
  "member_ports": [
    {
      "uuid": "2d3c9adc-f70d-11e8-b145-005056bb5b8e",
      "name": "e0f",
      "node": {
        "name": "user-cluster-01"
      }
    },
    {
      "uuid": "2d40b097-f70d-11e8-b145-005056bb5b8e",
      "name": "e0g",
      "node": {
        "name": "user-cluster-01"
      }
    },
    {
      "uuid": "2d46d01e-f70d-11e8-b145-005056bb5b8e",
      "name": "e0h",
      "node": {
        "name": "user-cluster-01"
      }
    }
  ]
}
]
}

```

Creating a LAG (ifgrp) port in a specific broadcast domain

The following output displays the record returned after the creation of a LAG port with "e0f", "e0g" and "e0h" as member ports. Also, the LAG port is added to the "Default" broadcast domain in the "Default" IPspace.

```

# The API:
/api/network/ethernet/ports

# The call:
curl -X POST "https://<mgmt-
ip>/api/network/ethernet/ports?return_records=true" -H "accept:
application/json" -H "Content-Type: application/json" -d "{  \"type\":
\"lag\",  \"node\": { \"name\": \"user-cluster-01\" },  \"broadcast_domain\": { \"name\": \"Default\", \"ipspace\": { \"name\": \"

```

```

\"Default\" } }, \"enabled\": true, \"lag\": { \"mode\": \"singlemode\",
\"distribution_policy\": \"mac\", \"member_ports\": [ { \"name\": \"e0f\",
\"node\": { \"name\": \"user-cluster-01\" } }, { \"name\": \"e0g\",
\"node\": { \"name\": \"user-cluster-01\" } }, { \"name\": \"e0h\",
\"node\": { \"name\": \"user-cluster-01\" } } ] } }" -u admin:netapp1! -k

# The response:
{
  "num_records": 1,
  "records": [
    {
      "uuid": "1807772a-fa4d-11e8-a6d7-005056bb5b8e",
      "type": "lag",
      "node": {
        "uuid": "faa56898-f70c-11e8-b145-005056bb5b8e",
        "name": "user-cluster-01"
      },
      "broadcast_domain": {
        "uuid": "36434bec-f70d-11e8-b145-005056bb5b8e",
        "name": "Default",
        "ipspace": {
          "name": "Default"
        }
      },
      "enabled": true,
      "lag": {
        "mode": "singlemode",
        "distribution_policy": "mac",
        "member_ports": [
          {
            "uuid": "2d3c9adc-f70d-11e8-b145-005056bb5b8e",
            "name": "e0f",
            "node": {
              "name": "user-cluster-01"
            }
          },
          {
            "uuid": "2d40b097-f70d-11e8-b145-005056bb5b8e",
            "name": "e0g",
            "node": {
              "name": "user-cluster-01"
            }
          },
          {
            "uuid": "2d46d01e-f70d-11e8-b145-005056bb5b8e",
            "name": "e0h",
            "node": {
              "name": "user-cluster-01"
            }
          }
        ]
      }
    }
  ]
}

```

```
        "node": {
            "name": "user-cluster-01"
        }
    }
}
]
}
```

Updating ports

You can use the network ports PATCH API to update the attributes of ports.

Examples

Updating the broadcast domain of a port

The following PATCH request removes the port from the current broadcast domain and adds it to the specified broadcast domain.

```
# The API:
/api/network/ethernet/ports/{uuid}

# The call:
curl -X PATCH "https://<mgmt-ip>/api/network/ethernet/ports/6867efaf-d702-11e8-994f-005056bbc994" -H "accept: application/hal+json" -H "Content-Type: application/json" -d "{ \"broadcast_domain\": { \"name\": \"Default\", \"ipspace\": { \"name\": \"Default\" } } }"
```

Updating the admin status of a port

The following PATCH request brings the specified port down.

```
# The API:  
/api/network/ethernet/ports/{uuid}  
  
# The call:  
curl -X PATCH "https://<mgmt-ip>/api/network/ethernet/ports/51d3ab39-d86d-  
11e8-aca6-005056bbc994" -H "accept: application/hal+json" -H "Content-  
Type: application/json" -d "{ \"enabled\": \"false\" }"
```

Repairing a port

The following PATCH request repairs a port. Only ports that have reachability as "repairable" can be repaired. The "reachability" parameter cannot be patched in the same request as other parameters that might affect the target port's reachability status.

```
# The API:  
/api/network/ethernet/ports/{uuid}  
  
# The call:  
curl -X PATCH "https://<mgmt-ip>/api/network/ethernet/ports/51d3ab39-d86d-  
11e8-aca6-005056bbc994" -H "accept: application/hal+json" -H "Content-  
Type: application/json" -d "{ \"reachability\": \"ok\" }"
```

Deleting ports

You can use the network ports DELETE API to delete VLAN and LAG ports in the cluster. Note that physical ports cannot be deleted. Deleting a port also removes the port from the broadcast domain.

Example

Deleting a VLAN port

The network ports DELETE API is used to delete a VLAN port.

```

# The API:
/api/network/ethernet/ports/{uuid}

# The call:
curl -X DELETE "https://<mgmt-ip>/api/network/ethernet/ports/6867efaf-
d702-11e8-994f-005056bbc994" -H "accept: application/hal+json" -H
"Content-Type: application/json"

```

Retrieve ports

GET /network/ethernet/ports

Introduced In: 9.6

Retrieves a collection of ports (physical, VLAN and LAG) for an entire cluster.

Related ONTAP commands

- `network port show`
- `network port ifgrp show`
- `network port vlan show`

Parameters

| Name | Type | In | Required | Description |
|---|--------|-------|----------|--|
| <code>reachable_broadcast_domains.uuid</code> | string | query | False | Filter by <code>reachable_broadcast_domains.uuid</code> <ul style="list-style-type: none"> • Introduced in: 9.8 |
| <code>reachable_broadcast_domains.name</code> | string | query | False | Filter by <code>reachable_broadcast_domains.name</code> <ul style="list-style-type: none"> • Introduced in: 9.8 |

| Name | Type | In | Required | Description |
|--|---------|-------|----------|---|
| reachable_broadcast_domains.ipspace.name | string | query | False | Filter by reachable_broadcast_domains.ipspace.name <ul style="list-style-type: none"> • Introduced in: 9.8 |
| state | string | query | False | Filter by state |
| statistics.timestamp | string | query | False | Filter by statistics.timestamp <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.throughput_raw.write | integer | query | False | Filter by statistics.throughput_raw.write <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.throughput_raw.read | integer | query | False | Filter by statistics.throughput_raw.read <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.throughput_raw.total | integer | query | False | Filter by statistics.throughput_raw.total <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.device.receive_raw.packets | integer | query | False | Filter by statistics.device.receive_raw.packets <ul style="list-style-type: none"> • Introduced in: 9.8 |

| Name | Type | In | Required | Description |
|---|---------|-------|----------|--|
| statistics.device.receive_raw.discards | integer | query | False | Filter by statistics.device.receive_raw.discards <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.device.receive_raw.errors | integer | query | False | Filter by statistics.device.receive_raw.errors <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.device.timestamp | string | query | False | Filter by statistics.device.timestamp <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.device.transmit_raw.packets | integer | query | False | Filter by statistics.device.transmit_raw.packets <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.device.transmit_raw.discards | integer | query | False | Filter by statistics.device.transmit_raw.discards <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.device.transmit_raw.errors | integer | query | False | Filter by statistics.device.transmit_raw.errors <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.device.link_down_count_raw | integer | query | False | Filter by statistics.device.link_down_count_raw <ul style="list-style-type: none"> • Introduced in: 9.8 |

| Name | Type | In | Required | Description |
|--------------------------|---------|-------|----------|--|
| statistics.status | string | query | False | Filter by statistics.status <ul style="list-style-type: none"> • Introduced in: 9.8 |
| speed | integer | query | False | Filter by speed |
| reachability | string | query | False | Filter by reachability <ul style="list-style-type: none"> • Introduced in: 9.8 |
| mac_address | string | query | False | Filter by mac_address |
| enabled | boolean | query | False | Filter by enabled |
| node.name | string | query | False | Filter by node.name |
| node.uuid | string | query | False | Filter by node.uuid |
| mtu | integer | query | False | Filter by mtu |
| vlan.tag | integer | query | False | Filter by vlan.tag |
| vlan.base_port.name | string | query | False | Filter by vlan.base_port.name |
| vlan.base_port.node.name | string | query | False | Filter by vlan.base_port.node.name |
| vlan.base_port.uuid | string | query | False | Filter by vlan.base_port.uuid |
| broadcast_domain.uid | string | query | False | Filter by broadcast_domain.uid |
| broadcast_domain.name | string | query | False | Filter by broadcast_domain.name |

| Name | Type | In | Required | Description |
|-----------------------------------|--------|-------|----------|---|
| broadcast_domain.ip space.name | string | query | False | Filter by broadcast_domain.ip space.name |
| lag.active_ports.name | string | query | False | Filter by lag.active_ports.name |
| lag.active_ports.node.name | string | query | False | Filter by lag.active_ports.node.name |
| lag.active_ports.uuid | string | query | False | Filter by lag.active_ports.uuid |
| lag.mode | string | query | False | Filter by lag.mode |
| lag.member_ports.name | string | query | False | Filter by lag.member_ports.name |
| lag.member_ports.node.name | string | query | False | Filter by lag.member_ports.node.name |
| lag.member_ports.uuid | string | query | False | Filter by lag.member_ports.uuid |
| lag.distribution_policy | string | query | False | Filter by lag.distribution_policy |
| name | string | query | False | Filter by name |
| uuid | string | query | False | Filter by uuid |
| metric.timestamp | string | query | False | Filter by metric.timestamp <ul style="list-style-type: none"> • Introduced in: 9.8 |

| Name | Type | In | Required | Description |
|-------------------------|---------------|-------|----------|--|
| metric.duration | string | query | False | Filter by metric.duration <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.throughput.write | integer | query | False | Filter by metric.throughput.write <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.throughput.read | integer | query | False | Filter by metric.throughput.read <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.throughput.total | integer | query | False | Filter by metric.throughput.total <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.status | string | query | False | Filter by metric.status <ul style="list-style-type: none"> • Introduced in: 9.8 |
| type | string | query | False | Filter by type |
| fields | array[string] | query | False | Specify the fields to return. |
| max_records | integer | query | False | Limit the number of records returned. |
| return_records | boolean | query | False | The default is true for GET calls. When set to false, only the number of records is returned. <ul style="list-style-type: none"> • Default value: 1 |

| Name | Type | In | Required | Description |
|----------------|---------------|-------|----------|--|
| return_timeout | integer | query | False | <p>The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached.</p> <ul style="list-style-type: none"> • Default value: 1 • Max value: 120 • Min value: 0 |
| order_by | array[string] | query | False | Order results by specified fields and optional [asc] |

Response

Status: 200, Ok

| Name | Type | Description |
|-------------|-------------------------------|-------------|
| _links | _links | |
| num_records | integer | |
| records | array[port] | |

Example response

```
{  
  "_links": {  
    "next": {  
      "href": "/api/resourcelink"  
    },  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "records": [  
    {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "broadcast_domain": {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        }  
      },  
      "ipspace": {  
        "name": "ipspace1"  
      },  
      "name": "bd1",  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    },  
    "lag": {  
      "active_ports": [  
        {  
          "_links": {  
            "self": {  
              "href": "/api/resourcelink"  
            }  
          },  
          "name": "elb",  
          "node": {  
            "name": "node1"  
          },  
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        }  
      ],  
      "distribution_policy": "string",  
      "name": "elb",  
      "node": {  
        "name": "node1"  
      },  
      "ports": [  
        {  
          "name": "port1",  
          "node": {  
            "name": "node1"  
          },  
          "order": 1,  
          "port": 10000,  
          "type": "ethernet"  
        },  
        {  
          "name": "port2",  
          "node": {  
            "name": "node1"  
          },  
          "order": 2,  
          "port": 10001,  
          "type": "ethernet"  
        }  
      ]  
    }  
  ]  
}
```

```

"member_ports": [
  {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "elb",
    "node": {
      "name": "node1"
    },
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  }
],
"mode": "string"
},
"mac_address": "01:02:03:04:05:06",
"metric": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "duration": "PT15S",
  "status": "ok",
  "throughput": {
    "read": "200",
    "total": "1000",
    "write": "100"
  },
  "timestamp": "2017-01-25T11:20:13Z"
},
"mtu": "1500",
"name": "elb",
"node": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "node1",
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"reachability": "ok",
"reachable_broadcast_domains": [
  {

```

```

    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "ipspace": {
        "name": "ipspace1"
    },
    "name": "bd1",
    "uuid": "1cd8a442-86d1-11e0-aelc-123478563412"
}
],
"speed": "1000",
"state": "string",
"statistics": {
    "device": {
        "link_down_count_raw": "3",
        "receive_raw": {
            "discards": "100",
            "errors": "200",
            "packets": "500"
        },
        "timestamp": "2017-01-25T11:20:13Z",
        "transmit_raw": {
            "discards": "100",
            "errors": "200",
            "packets": "500"
        }
    },
    "status": "ok",
    "throughput_raw": {
        "read": "200",
        "total": "1000",
        "write": "100"
    },
    "timestamp": "2017-01-25T11:20:13Z"
},
"type": "string",
"uuid": "1cd8a442-86d1-11e0-aelc-123478563412",
"vlan": {
    "base_port": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "name": "bd1"
    }
}
]

```

```

        "name": "elb",
        "node": {
            "name": "node1"
        },
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "tag": "100"
}
]
}

```

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 | |

ipspace

| Name | Type | Description |
|------|--------|--|
| name | string | Name of the broadcast domain's IPspace |

broadcast_domain

Broadcast domain UUID along with a readable name. Either the UUID or both names may be provided on input.

| Name | Type | Description |
|---------|---------|---|
| _links | _links | |
| ipspace | ipspace | |
| name | string | Name of the broadcast domain, scoped to its IPspace |
| uuid | string | Broadcast domain UUID |

node

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

active_ports

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

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| node | node | |
| uuid | string | |

member_ports

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

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| node | node | |
| uuid | string | |

lag

| Name | Type | Description |
|---------------------|---------------------------------------|---|
| active_ports | array[active_ports] | Active ports of a LAG (ifgrp). (Some member ports may be inactive.) |
| distribution_policy | string | Policy for mapping flows to ports for outbound packets through a LAG (ifgrp). |
| member_ports | array[member_ports] | |
| mode | string | Determines how the ports interact with the switch. |

throughput

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

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

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

metric

The most recent sample of I/O metrics for the port.

| 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: |
| 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 port object. |

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

node

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| uuid | string | |

reachable_broadcast_domains

Broadcast domain UUID along with a readable name. Either the UUID or both names may be provided on input.

| Name | Type | Description |
|---------|-------------------------|---|
| _links | _links | |
| ipspace | ipspace | |
| name | string | Name of the broadcast domain, scoped to its IPspace |
| uuid | string | Broadcast domain UUID |

receive_raw

Packet receive counters for the Ethernet port.

| Name | Type | Description |
|----------|---------|------------------------------------|
| discards | integer | Total number of discarded packets. |
| errors | integer | Number of packet errors. |
| packets | integer | Total packet count. |

transmit_raw

Packet transmit counters for the Ethernet port.

| Name | Type | Description |
|----------|---------|------------------------------------|
| discards | integer | Total number of discarded packets. |
| errors | integer | Number of packet errors. |
| packets | integer | Total packet count. |

device

Device-related counters for the port object. These counters are applicable at the lowest layer of the networking stack. These values can be used to calculate both transmit and receive packet and error rates by comparing two samples taken at different times and calculating the increase in counter value divided by the elapsed time between the two samples.

| Name | Type | Description |
|---------------------|------------------------------|--|
| link_down_count_raw | integer | The number of link state changes from up to down seen on the device. |
| receive_raw | receive_raw | Packet receive counters for the Ethernet port. |
| timestamp | string | The timestamp when the device specific counters were collected. |
| transmit_raw | transmit_raw | Packet transmit counters for the Ethernet port. |

throughput_raw

Throughput bytes observed at the port object. This can 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

The real time I/O statistics for the port.

| Name | Type | Description |
|----------------|----------------|---|
| device | device | Device-related counters for the port object. These counters are applicable at the lowest layer of the networking stack. These values can be used to calculate both transmit and receive packet and error rates by comparing two samples taken at different times and calculating the increase in counter value divided by the elapsed time between the two samples. |
| 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_raw | throughput_raw | Throughput bytes observed at the port object. This can be used along with delta time to calculate the rate of throughput bytes per unit of time. |
| timestamp | string | The timestamp of the throughput_raw performance data. |

base_port

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

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| node | node | |
| uuid | string | |

vlan

| Name | Type | Description |
|-----------|---------------------------|--|
| base_port | base_port | Port UUID along with readable names. Either the UUID or both names may be supplied on input. |
| tag | integer | VLAN ID |

port

| Name | Type | Description |
|------------------|----------------------------------|---|
| _links | _links | |
| broadcast_domain | broadcast_domain | Broadcast domain UUID along with a readable name. Either the UUID or both names may be provided on input. |
| enabled | boolean | |
| lag | lag | |
| mac_address | string | |
| metric | metric | The most recent sample of I/O metrics for the port. |
| mtu | integer | MTU of the port in bytes. Set by broadcast domain. |
| name | string | Portname, such as e0a, e1b-100 (VLAN on ethernet), a0c (LAG/ifgrp), a0d-200 (vlan on LAG/ifgrp) |
| node | node | |

| Name | Type | Description |
|-----------------------------|------------------------------------|---|
| reachability | string | Reachability status of the port. Enum value "ok" is the only acceptable value for a PATCH request to repair a port. |
| reachable_broadcast_domains | array[reachable_broadcast_domains] | Reachable broadcast domains. |
| speed | integer | Link speed in Mbps |
| state | string | Operational state of the port. |
| statistics | statistics | The real time I/O statistics for the port. |
| type | string | Type of physical or virtual port |
| uuid | string | Port UUID |
| vlan | vlan | |

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 a new VLAN or LAG

POST /network/ethernet/ports

Introduced In: 9.6

Creates a new VLAN (such as node1:e0a-100) or LAG (ifgrp, such as node2:a0a).

Required properties

- node - Node the port will be created on.
- type - Defines if a VLAN or LAG will be created:
 - VLAN
 - `vlan.base_port` - Physical port or LAG the VLAN will be created on.
 - `vlan.tag` - Tag used to identify VLAN on the base port.
 - LAG
 - `lag.mode` - Policy for the LAG that will be created.
 - `lag.distribution_policy` - Indicates how the packets are distributed between ports.
 - `lag.member_ports` - Set of ports the LAG consists of.

Optional properties

- `broadcast_domain` - The layer-2 broadcast domain the port is associated with. The port will be placed in a broadcast domain if it is not specified. It may take several minutes for the broadcast domain to be assigned. During that period the port cannot host interfaces.

Related ONTAP commands

- `network port ifgrp create`
- `network port vlan create`

Parameters

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

Request Body

| Name | Type | Description |
|-------------------------------|-------------------------------|---|
| <code>broadcast_domain</code> | <code>broadcast_domain</code> | Broadcast domain UUID along with a readable name. Either the UUID or both names may be provided on input. |

| Name | Type | Description |
|-----------------------------|--|---|
| enabled | boolean | |
| lag | lag | |
| mac_address | string | |
| mtu | integer | MTU of the port in bytes. Set by broadcast domain. |
| name | string | Portname, such as e0a, e1b-100 (VLAN on ethernet), a0c (LAG/ifgrp), a0d-200 (vlan on LAG/ifgrp) |
| node | node | |
| reachable_broadcast_domains | array[reachable_broadcast_domain s] | Reachable broadcast domains. |
| speed | integer | Link speed in Mbps |
| state | string | Operational state of the port. |
| type | string | Type of physical or virtual port |
| uuid | string | Port UUID |
| vlan | vlan | |

Example request

```
{  
  "broadcast_domain": {  
    "ipspace": {  
      "name": "ipspacel"  
    },  
    "name": "bd1",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "lag": {  
    "active_ports": [  
      {  
        "name": "elb",  
        "node": {  
          "name": "node1"  
        },  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
      }  
    ],  
    "distribution_policy": "string",  
    "member_ports": [  
      {  
        "name": "elb",  
        "node": {  
          "name": "node1"  
        },  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
      }  
    ],  
    "mode": "string"  
  },  
  "mac_address": "01:02:03:04:05:06",  
  "mtu": "1500",  
  "name": "elb",  
  "node": {  
    "name": "node1",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "reachable_broadcast_domains": [  
    {  
      "ipspace": {  
        "name": "ipspacel"  
      },  
      "name": "bd1",  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    }  
  ]  
}
```

```

        }
    ],
    "speed": "1000",
    "state": "string",
    "type": "string",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",
    "vlan": {
        "base_port": {
            "name": "elb",
            "node": {
                "name": "node1"
            },
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
        },
        "tag": "100"
    }
}

```

Response

Status: 201, Created

| Name | Type | Description |
|-------------|-------------|-------------|
| num_records | integer | |
| records | array[port] | |

Example response

```
{  
  "records": [  
    {  
      "broadcast_domain": {  
        "ipspace": {  
          "name": "ipspacel"  
        },  
        "name": "bd1",  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
      },  
      "lag": {  
        "active_ports": [  
          {  
            "name": "elb",  
            "node": {  
              "name": "node1"  
            },  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
          }  
        ],  
        "distribution_policy": "string",  
        "member_ports": [  
          {  
            "name": "elb",  
            "node": {  
              "name": "node1"  
            },  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
          }  
        ],  
        "mode": "string"  
      },  
      "mac_address": "01:02:03:04:05:06",  
      "mtu": "1500",  
      "name": "elb",  
      "node": {  
        "name": "node1",  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
      },  
      "reachable_broadcast_domains": [  
        {  
          "ipspace": {  
            "name": "ipspacel"  
          },  
          "name": "bd1",  
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        }  
      ]  
    }  
  ]  
}
```

```

        "name": "bd1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
],
"speed": "1000",
"state": "string",
"type": "string",
"uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",
"vlan": {
    "base_port": {
        "name": "e1b",
        "node": {
            "name": "node1"
        },
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "tag": "100"
}
}
]
}

```

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|--|
| 1376361 | Port is already a member of a LAG. |
| 1966189 | Port is the home port or current port of an interface. |
| 1967083 | The specified type is not valid. |
| 1967084 | The specified node UUID is not valid. |
| 1967085 | The specified node name is not valid. |
| 1967086 | Node name and UUID must match if both are provided. |
| 1967087 | The specified broadcast domain UUID is not valid. |
| 1967088 | The specified broadcast domain name does not exist in the specified IPspace. |
| 1967089 | The specified broadcast domain UUID, name, and IPspace name do not match. |

| Error Code | Description |
|------------|--|
| 1967090 | The specified VLAN base port UUID is not valid. |
| 1967091 | The specified VLAN base port name and node name are not valid. |
| 1967092 | The specified node does not match the node specified for the VLAN base port. |
| 1967093 | The specified VLAN base port UUID, name, and VLAN base port node name do not match. |
| 1967094 | The specified LAG member port UUID is not valid. |
| 1967095 | The specified LAG member port name and node name combination is not valid. |
| 1967096 | The specified node does not match the specified LAG member port node. |
| 1967097 | The specified LAG member ports UUID, name, and node name do not match. |
| 1967098 | VLAN POST operation has failed because admin status could not be set for the specified port. |
| 1967099 | Partial success of the VLAN POST operation. Verify the state of the created VLAN for more information. |
| 1967100 | LAG POST operation failed because admin status could not be set. |
| 1967101 | Partial success of the LAG POST operation. Verify the state of the created LAG for more information. |
| 1967102 | POST operation might have left the configuration in an inconsistent state. Check the configuration. |
| 1967148 | Failure to remove port from broadcast domain. |
| 1967149 | Failure to add port to broadcast domain. |
| 1967175 | VLANs cannot be created on ports in the Cluster IPspace. |

Definitions

See Definitions

href

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

_links

ipspace

| Name | Type | Description |
|------|--------|--|
| name | string | Name of the broadcast domain's IPspace |

broadcast_domain

Broadcast domain UUID along with a readable name. Either the UUID or both names may be provided on input.

| Name | Type | Description |
|---------|---------|---|
| ipspace | ipspace | |
| name | string | Name of the broadcast domain, scoped to its IPspace |
| uuid | string | Broadcast domain UUID |

node

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

active_ports

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

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| node | node | |
| uuid | string | |

member_ports

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

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| node | node | |
| uuid | string | |

lag

| Name | Type | Description |
|---------------------|---------------------|---|
| active_ports | array[active_ports] | Active ports of a LAG (ifgrp). (Some member ports may be inactive.) |
| distribution_policy | string | Policy for mapping flows to ports for outbound packets through a LAG (ifgrp). |
| member_ports | array[member_ports] | |
| mode | string | Determines how the ports interact with the switch. |

throughput

The rate of throughput bytes per second observed at the port 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

The most recent sample of I/O metrics for the port.

| 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: |
| 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 port object. |
| timestamp | string | The timestamp of the performance data. |

node

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| uuid | string | |

reachable_broadcast_domains

Broadcast domain UUID along with a readable name. Either the UUID or both names may be provided on

input.

| Name | Type | Description |
|---------|---------|---|
| ipspace | ipspace | |
| name | string | Name of the broadcast domain, scoped to its IPspace |
| uuid | string | Broadcast domain UUID |

receive_raw

Packet receive counters for the Ethernet port.

| Name | Type | Description |
|----------|---------|------------------------------------|
| discards | integer | Total number of discarded packets. |
| errors | integer | Number of packet errors. |
| packets | integer | Total packet count. |

transmit_raw

Packet transmit counters for the Ethernet port.

| Name | Type | Description |
|----------|---------|------------------------------------|
| discards | integer | Total number of discarded packets. |
| errors | integer | Number of packet errors. |
| packets | integer | Total packet count. |

device

Device-related counters for the port object. These counters are applicable at the lowest layer of the networking stack. These values can be used to calculate both transmit and receive packet and error rates by comparing two samples taken at different times and calculating the increase in counter value divided by the elapsed time between the two samples.

| Name | Type | Description |
|---------------------|---------|--|
| link_down_count_raw | integer | The number of link state changes from up to down seen on the device. |

| Name | Type | Description |
|-----------|--------|---|
| timestamp | string | The timestamp when the device specific counters were collected. |

throughput_raw

Throughput bytes observed at the port object. This can 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

The real time I/O statistics for the port.

| Name | Type | Description |
|----------------|--------------------------------|--|
| status | string | <p>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".</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 port object. This can be used along with delta time to calculate the rate of throughput bytes per unit of time. |
| timestamp | string | The timestamp of the throughput_raw performance data. |

base_port

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

| Name | Type | Description |
|------|----------------------|-------------|
| name | string | |
| node | node | |
| uuid | string | |

vlan

| Name | Type | Description |
|-----------|-----------|--|
| base_port | base_port | Port UUID along with readable names. Either the UUID or both names may be supplied on input. |
| tag | integer | VLAN ID |

port

| Name | Type | Description |
|-----------------------------|------------------------------------|---|
| broadcast_domain | broadcast_domain | Broadcast domain UUID along with a readable name. Either the UUID or both names may be provided on input. |
| enabled | boolean | |
| lag | lag | |
| mac_address | string | |
| mtu | integer | MTU of the port in bytes. Set by broadcast domain. |
| name | string | Portname, such as e0a, e1b-100 (VLAN on ethernet), a0c (LAG/ifgrp), a0d-200 (vlan on LAG/ifgrp) |
| node | node | |
| reachable_broadcast_domains | array[reachable_broadcast_domains] | Reachable broadcast domains. |
| speed | integer | Link speed in Mbps |
| state | string | Operational state of the port. |
| type | string | Type of physical or virtual port |
| uuid | string | Port UUID |
| vlan | vlan | |

error_arguments

| Name | Type | Description |
|------|--------|---------------|
| code | string | Argument code |

| Name | Type | Description |
|-----------|------------------------|---|
| 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 a VLAN or LAG

DELETE /network/ethernet/ports/{uuid}

Introduced In: 9.6

Deletes a VLAN or LAG.

Related ONTAP commands

- `network port ifgrp delete`
- `network port vlan delete`

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|-------------|
| uuid | string | path | True | Port UUID |

Response

Status: 200, Ok

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|--|
| 1376858 | Port already has an interface bound. |
| 1966189 | Port is the home port or current port of an interface. |

| 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 a physical port, VLAN, or LAG details

GET /network/ethernet/ports/{uuid}

Introduced In: 9.6

Retrieves the details of a physical port, VLAN, or LAG.

Related ONTAP commands

- `network port show`
- `network port ifgrp show`
- `network port vlan show`

Parameters

| Name | Type | In | Required | Description |
|--------|---------------|-------|----------|-------------------------------|
| uuid | string | path | True | Port UUID |
| fields | array[string] | query | False | Specify the fields to return. |

Response

Status: 200, Ok

| Name | Type | Description |
|-----------------------------|------------------------------------|---|
| _links | _links | |
| broadcast_domain | broadcast_domain | Broadcast domain UUID along with a readable name. Either the UUID or both names may be provided on input. |
| enabled | boolean | |
| lag | lag | |
| mac_address | string | |
| metric | metric | The most recent sample of I/O metrics for the port. |
| mtu | integer | MTU of the port in bytes. Set by broadcast domain. |
| name | string | Portname, such as e0a, e1b-100 (VLAN on ethernet), a0c (LAG/ifgrp), a0d-200 (vlan on LAG/ifgrp) |
| node | node | |
| reachability | string | Reachability status of the port. Enum value "ok" is the only acceptable value for a PATCH request to repair a port. |
| reachable_broadcast_domains | array[reachable_broadcast_domains] | Reachable broadcast domains. |
| speed | integer | Link speed in Mbps |
| state | string | Operational state of the port. |
| statistics | statistics | The real time I/O statistics for the port. |
| type | string | Type of physical or virtual port |
| uuid | string | Port UUID |

| Name | Type | Description |
|------|----------------------|-------------|
| vlan | vlan | |

Example response

```
{  
  "_links": {  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "broadcast_domain": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    },  
    "ipspace": {  
      "name": "ipspace1"  
    },  
    "name": "bd1",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "lag": {  
    "active_ports": [  
      {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        },  
        "name": "e1b",  
        "node": {  
          "name": "node1"  
        },  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
      }  
    ],  
    "distribution_policy": "string",  
    "member_ports": [  
      {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        },  
        "name": "e1b",  
        "node": {  
          "name": "node1"  
        }  
      }  
    ]  
  }  
}
```

```
        },
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
],
"mode": "string"
},
"mac_address": "01:02:03:04:05:06",
"metric": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "duration": "PT15S",
    "status": "ok",
    "throughput": {
        "read": "200",
        "total": "1000",
        "write": "100"
    },
    "timestamp": "2017-01-25T11:20:13Z"
},
"mtu": "1500",
"name": "elb",
"node": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "name": "node1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"reachability": "ok",
"reachable_broadcast_domains": [
{
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "ipspace": {
        "name": "ipspacel"
    },
    "name": "bd1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
```

```
        }
    ],
    "speed": "1000",
    "state": "string",
    "statistics": {
        "device": {
            "link_down_count_raw": "3",
            "receive_raw": {
                "discards": "100",
                "errors": "200",
                "packets": "500"
            },
            "timestamp": "2017-01-25T11:20:13Z",
            "transmit_raw": {
                "discards": "100",
                "errors": "200",
                "packets": "500"
            }
        },
        "status": "ok",
        "throughput_raw": {
            "read": "200",
            "total": "1000",
            "write": "100"
        },
        "timestamp": "2017-01-25T11:20:13Z"
    },
    "type": "string",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",
    "vlan": {
        "base_port": {
            "_links": {
                "self": {
                    "href": "/api/resourcelink"
                }
            },
            "name": "elb",
            "node": {
                "name": "node1"
            },
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
        },
        "tag": "100"
    }
}
```

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 | |

ipspace

| Name | Type | Description |
|------|--------|--|
| name | string | Name of the broadcast domain's IPspace |

broadcast_domain

Broadcast domain UUID along with a readable name. Either the UUID or both names may be provided on input.

| Name | Type | Description |
|---------|---------|---|
| _links | _links | |
| ipspace | ipspace | |
| name | string | Name of the broadcast domain, scoped to its IPspace |
| uuid | string | Broadcast domain UUID |

node

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

active_ports

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

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

| Name | Type | Description |
|------|----------------------|-------------|
| name | string | |
| node | node | |
| uuid | string | |

member_ports

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

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| node | node | |
| uuid | string | |

lag

| Name | Type | Description |
|---------------------|---------------------------------------|---|
| active_ports | array[active_ports] | Active ports of a LAG (ifgrp). (Some member ports may be inactive.) |
| distribution_policy | string | Policy for mapping flows to ports for outbound packets through a LAG (ifgrp). |
| member_ports | array[member_ports] | |
| mode | string | Determines how the ports interact with the switch. |

throughput

The rate of throughput bytes per second observed at the port 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

The most recent sample of I/O metrics for the port.

| 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: |
| 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 port object. |
| timestamp | string | The timestamp of the performance data. |

node

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

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| uuid | string | |

reachable_broadcast_domains

Broadcast domain UUID along with a readable name. Either the UUID or both names may be provided on input.

| Name | Type | Description |
|---------|-------------------------|---|
| _links | _links | |
| ipspace | ipspace | |
| name | string | Name of the broadcast domain, scoped to its IPspace |
| uuid | string | Broadcast domain UUID |

receive_raw

Packet receive counters for the Ethernet port.

| Name | Type | Description |
|----------|---------|------------------------------------|
| discards | integer | Total number of discarded packets. |
| errors | integer | Number of packet errors. |
| packets | integer | Total packet count. |

transmit_raw

Packet transmit counters for the Ethernet port.

| Name | Type | Description |
|----------|---------|------------------------------------|
| discards | integer | Total number of discarded packets. |
| errors | integer | Number of packet errors. |
| packets | integer | Total packet count. |

device

Device-related counters for the port object. These counters are applicable at the lowest layer of the

networking stack. These values can be used to calculate both transmit and receive packet and error rates by comparing two samples taken at different times and calculating the increase in counter value divided by the elapsed time between the two samples.

| Name | Type | Description |
|---------------------|------------------------------|--|
| link_down_count_raw | integer | The number of link state changes from up to down seen on the device. |
| receive_raw | receive_raw | Packet receive counters for the Ethernet port. |
| timestamp | string | The timestamp when the device specific counters were collected. |
| transmit_raw | transmit_raw | Packet transmit counters for the Ethernet port. |

throughput_raw

Throughput bytes observed at the port object. This can 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

The real time I/O statistics for the port.

| Name | Type | Description |
|----------------|--------------------------------|---|
| device | device | Device-related counters for the port object. These counters are applicable at the lowest layer of the networking stack. These values can be used to calculate both transmit and receive packet and error rates by comparing two samples taken at different times and calculating the increase in counter value divided by the elapsed time between the two samples. |
| 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_raw | throughput_raw | Throughput bytes observed at the port object. This can be used along with delta time to calculate the rate of throughput bytes per unit of time. |
| timestamp | string | The timestamp of the throughput_raw performance data. |

base_port

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

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| node | node | |
| uuid | string | |

vlan

| Name | Type | Description |
|-----------|---------------------------|--|
| base_port | base_port | Port UUID along with readable names. Either the UUID or both names may be supplied on input. |
| tag | integer | VLAN ID |

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 a port

PATCH /network/ethernet/ports/{uuid}

Introduced In: 9.6

Updates a port.

Related ONTAP commands

- `network port broadcast-domain add-ports`
- `network port broadcast-domain remove-ports`
- `network port ifgrp modify`
- `network port modify`
- `network port vlan modify`

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|-------------|
| uuid | string | path | True | Port UUID |

Request Body

| Name | Type | Description |
|-----------------------------|--|---|
| broadcast_domain | broadcast_domain | Broadcast domain UUID along with a readable name. Either the UUID or both names may be provided on input. |
| enabled | boolean | |
| lag | lag | |
| mac_address | string | |
| mtu | integer | MTU of the port in bytes. Set by broadcast domain. |
| name | string | Portname, such as e0a, e1b-100 (VLAN on ethernet), a0c (LAG/ifgrp), a0d-200 (vlan on LAG/ifgrp) |
| reachability | string | Reachability status of the port. Enum value "ok" is the only acceptable value for a PATCH request to repair a port. |
| reachable_broadcast_domains | array[reachable_broadcast_domain s] | Reachable broadcast domains. |
| speed | integer | Link speed in Mbps |
| state | string | Operational state of the port. |

| Name | Type | Description |
|------|--------|-------------|
| uuid | string | Port UUID |

Example request

```
{  
    "broadcast_domain": {  
        "ipspace": {  
            "name": "ipspace1"  
        },  
        "name": "bd1",  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    },  
    "lag": {  
        "active_ports": [  
            {  
                "name": "e1b",  
                "node": {  
                    "name": "node1"  
                },  
                "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
            }  
        ],  
        "member_ports": [  
            {  
                "name": "e1b",  
                "node": {  
                    "name": "node1"  
                },  
                "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
            }  
        ]  
    },  
    "mac_address": "01:02:03:04:05:06",  
    "mtu": "1500",  
    "name": "e1b",  
    "reachability": "ok",  
    "reachable_broadcast_domains": [  
        {  
            "ipspace": {  
                "name": "ipspace1"  
            },  
            "name": "bd1",  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        }  
    ],  
    "speed": "1000",  
    "state": "string",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
}
```

Response

Status: 200, Ok

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|---|
| 1376361 | Port is already a member of a LAG. |
| 1377562 | Port cannot be used because it is currently the home port or current port of an interface. |
| 1377563 | Port is already a member of a LAG. |
| 1967087 | The specified broadcast domain UUID is not valid. |
| 1967088 | The specified broadcast domain name does not exist in the specified IPspace. |
| 1967089 | The specified broadcast domain UUID, name and IPspace name do not match. |
| 1967094 | The specified LAG member port UUID is not valid. |
| 1967095 | The specified LAG member port name and node name combination is not valid. |
| 1967096 | The specified node does not match the specified LAG member port node. |
| 1967097 | The specified LAG member ports UUID, name, and node name do not match. |
| 1967148 | Failure to remove port from broadcast domain. |
| 1967149 | Failure to add port to broadcast domain. |
| 1967184 | The reachability parameter cannot be patched in the same request as other parameters that might affect the target port's reachability status. |
| 1967185 | The port cannot be repaired because the port is deemed as non-repairable. |
| 1967186 | Invalid value for the reachability parameter. |
| 1967580 | This command is not supported as the effective cluster version is earlier than 9.8. |
| 1967582 | The reachability parameter is not supported on this cluster. |

Definitions

See Definitions

href

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

_links

ipspace

| Name | Type | Description |
|------|--------|--|
| name | string | Name of the broadcast domain's IPspace |

broadcast_domain

Broadcast domain UUID along with a readable name. Either the UUID or both names may be provided on input.

| Name | Type | Description |
|---------|---------|---|
| ipspace | ipspace | |
| name | string | Name of the broadcast domain, scoped to its IPspace |
| uuid | string | Broadcast domain UUID |

node

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

active_ports

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

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| node | node | |
| uuid | string | |

member_ports

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

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| node | node | |
| uuid | string | |

lag

| Name | Type | Description |
|--------------|---------------------|---|
| active_ports | array[active_ports] | Active ports of a LAG (ifgrp). (Some member ports may be inactive.) |
| member_ports | array[member_ports] | |

throughput

The rate of throughput bytes per second observed at the port 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

The most recent sample of I/O metrics for the port.

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

| Name | Type | Description |
|------------|------------|--|
| status | string | <p>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".</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 | throughput | The rate of throughput bytes per second observed at the port object. |
| timestamp | string | The timestamp of the performance data. |

node

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| uuid | string | |

reachable_broadcast_domains

Broadcast domain UUID along with a readable name. Either the UUID or both names may be provided on input.

| Name | Type | Description |
|---------|---------|-------------|
| ipspace | ipspace | |

| Name | Type | Description |
|------|--------|---|
| name | string | Name of the broadcast domain, scoped to its IPspace |
| uuid | string | Broadcast domain UUID |

receive_raw

Packet receive counters for the Ethernet port.

| Name | Type | Description |
|----------|---------|------------------------------------|
| discards | integer | Total number of discarded packets. |
| errors | integer | Number of packet errors. |
| packets | integer | Total packet count. |

transmit_raw

Packet transmit counters for the Ethernet port.

| Name | Type | Description |
|----------|---------|------------------------------------|
| discards | integer | Total number of discarded packets. |
| errors | integer | Number of packet errors. |
| packets | integer | Total packet count. |

device

Device-related counters for the port object. These counters are applicable at the lowest layer of the networking stack. These values can be used to calculate both transmit and receive packet and error rates by comparing two samples taken at different times and calculating the increase in counter value divided by the elapsed time between the two samples.

| Name | Type | Description |
|---------------------|---------|--|
| link_down_count_raw | integer | The number of link state changes from up to down seen on the device. |
| timestamp | string | The timestamp when the device specific counters were collected. |

throughput_raw

Throughput bytes observed at the port object. This can 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

The real time I/O statistics for the port.

| Name | Type | Description |
|--------|--------|---|
| 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. |

| Name | Type | Description |
|----------------|--------------------------------|--|
| throughput_raw | throughput_raw | Throughput bytes observed at the port object. This can be used along with delta time to calculate the rate of throughput bytes per unit of time. |
| timestamp | string | The timestamp of the throughput_raw performance data. |

base_port

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

| Name | Type | Description |
|------|----------------------|-------------|
| name | string | |
| node | node | |
| uuid | string | |

vlan

| Name | Type | Description |
|-----------|---------------------------|--|
| base_port | base_port | Port UUID along with readable names. Either the UUID or both names may be supplied on input. |
| tag | integer | VLAN ID |

port

| Name | Type | Description |
|------------------|----------------------------------|---|
| broadcast_domain | broadcast_domain | Broadcast domain UUID along with a readable name. Either the UUID or both names may be provided on input. |
| enabled | boolean | |
| lag | lag | |
| mac_address | string | |
| mtu | integer | MTU of the port in bytes. Set by broadcast domain. |

| Name | Type | Description |
|-----------------------------|------------------------------------|---|
| name | string | Portname, such as e0a, e1b-100 (VLAN on ethernet), a0c (LAG/ifgrp), a0d-200 (vlan on LAG/ifgrp) |
| reachability | string | Reachability status of the port. Enum value "ok" is the only acceptable value for a PATCH request to repair a port. |
| reachable_broadcast_domains | array[reachable_broadcast_domains] | Reachable broadcast domains. |
| speed | integer | Link speed in Mbps |
| state | string | Operational state of the port. |
| uuid | string | Port UUID |

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 historical port performance metrics

GET /network/ethernet/ports/{uuid}/metrics

Introduced In: 9.8

Retrieves historical performance metrics for a port.

Parameters

| Name | Type | In | Required | Description |
|------------------|---------|-------|----------|--------------------------------|
| timestamp | string | query | False | Filter by timestamp |
| duration | string | query | False | Filter by duration |
| 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 |
| status | string | query | False | Filter by status |
| uuid | string | path | True | Unique identifier of the port. |

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

| Name | Type | In | Required | Description |
|--|----------------|---------|----------|--|
| return_timeout | integer | query | False | <p>The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached.</p> <ul style="list-style-type: none"> • Default value: 1 • Max value: 120 • Min value: 0 |
| fields | array[string] | query | False | Specify the fields to return. |
| max_records | integer | query | False | Limit the number of records returned. |
| order_by | array[string] | query | False | Order results by specified fields and optional [asc] |
| desc] direction. Default direction is 'asc' for ascending. | return_records | boolean | query | False |

Response

Status: 200, Ok

| Name | Type | Description |
|-------------|------------------------|-------------------|
| _links | _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",  
      "status": "ok",  
      "throughput": {  
        "read": "200",  
        "total": "1000",  
        "write": "100"  
      },  
      "timestamp": "2017-01-25T11:20:13Z"  
    }  
  ]  
}
```

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 | |

throughput

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

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

records

Throughput performance for the Ethernet port.

| 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: |

| Name | Type | Description |
|------------|------------|--|
| status | string | <p>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".</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 | throughput | The rate of throughput bytes per second observed at the port 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 |

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

Retrieve network Ethernet switch ports

Network Ethernet switch ports endpoint overview

Overview

This API can be used to get the port information for an ethernet switch used in a cluster or storage networks. This API supports GET only. The GET operation returns a list of ports with status and configuration information.

Examples

Retrieving the ports for ethernet switches

The following example retrieves the ethernet switch ports for all the ethernet switches used for cluster and/or storage networks. Note that if the *fields=** parameter is not specified, the fields in-octets, in-errors, in-discards, out-octets, out-errors, out-discards, interface-number, unique-name, mac-address are not returned. Filters can be added on the fields to limit the results.

```
# The API:
GET /network/ethernet/switch/ports

# The call:
curl -X GET "https://<mgmt-
ip>/api/network/ethernet/switch/ports?return_records=true" -H "accept:
application/json" -H "Content-Type: application/hal+json"

# The response: (abbreviated output due to length, shows a port connected
to a node, a port with no remote connection and a port connected to shelf)
{
  "records": [
    {
      "switch": {
        "name": "RTP-CS01-510R11 (FOC22092K12)",
        "_links": {
          "self": {
            "href": "/api/network/ethernet/switches/RTP-CS01-
510R11 (FOC22092K12)"
          }
        }
      }
    }
  ]
}
```

```

} ,
"identity": {
  "name": "Ethernet1/1",
  "index": 436207616,
  "number": 1,
  "type": "ethernetcsmacd"
},
"mtu": 9216,
"duplex_type": "full_duplex",
"speed": 100000,
"configured": "up",
"state": "up",
"isl": false,
"statistics": {
  "receive_raw": {
    "packets": 1616467751,
    "errors": 0,
    "discards": 0
  },
  "transmit_raw": {
    "packets": 206717534,
    "errors": 0,
    "discards": 0
  }
},
"remote_port": {
  "mtu": 9000,
  "name": "e3a",
  "device": {
    "node": {
      "name": "stiA400-311",
      "uuid": "54c0f036-8a3a-11ea-893d-00a098fd726d",
      "_links": {
        "self": {
          "href": "/api/cluster/nodes/54c0f036-8a3a-11ea-893d-00a098fd726d"
        }
      }
    }
  }
},
"mac_address": "00be75ae2ad4",
"vlan_id": [
  1,
  17,
  18,

```

```

92
  ],
  "_links": {
    "self": {
      "href": "/api/network/ethernet/switch/ports/RTP-CS01-
510R11%28FOC22092K12%29/Ethernet1%2F1/436207616"
    }
  }
},
{
  "switch": {
    "name": "RTP-CS01-510R11 (FOC22092K12)",
    "_links": {
      "self": {
        "href": "/api/network/ethernet/switches/RTP-CS01-
510R11 (FOC22092K12)"
      }
    }
  },
  "identity": {
    "name": "Ethernet1/11",
    "index": 436212736,
    "number": 11,
    "type": "ethernetcsmacl"
  },
  "mtu": 9216,
  "duplex_type": "unknown",
  "speed": 100000,
  "configured": "up",
  "state": "down",
  "isl": false,
  "statistics": {
    "receive_raw": {
      "packets": 0,
      "errors": 0,
      "discards": 0
    },
    "transmit_raw": {
      "packets": 0,
      "errors": 0,
      "discards": 0
    }
  },
  "mac_address": "00be75ae2afc",
  "vlan_id": [
    1,

```

```

17,
18,
92
],
"_links": {
  "self": {
    "href": "/api/network/ethernet/switch/ports/RTP-CS01-
510R11%28FOC22092K12%29/Ethernet1%2F11/436212736"
  }
}
},
{
  "switch": {
    "name": "RTP-SS01-510R10 (FOC22170DFR)",
    "_links": {
      "self": {
        "href": "/api/network/ethernet/switches/RTP-SS01-
510R10 (FOC22170DFR)"
      }
    }
  },
  "identity": {
    "name": "Ethernet1/10",
    "index": 436212224,
    "number": 10,
    "type": "ethernetcsmacd"
  },
  "mtu": 9216,
  "duplex_type": "full_duplex",
  "speed": 100000,
  "configured": "up",
  "state": "up",
  "isl": false,
  "statistics": {
    "receive_raw": {
      "packets": 332013844,
      "errors": 0,
      "discards": 0
    },
    "transmit_raw": {
      "packets": 2429595607,
      "errors": 0,
      "discards": 0
    }
  },
  "remote_port": {

```

```

    "mtu": 9000,
    "name": "e0a",
    "device": {
        "shelf": {
            "name": "SHFFG1828000004:B",
            "uid": "12439000444923584512",
            "_links": {
                "self": {
                    "href": "/api/storage/shelves/12439000444923584512"
                }
            }
        }
    },
    "mac_address": "00fcbaead548",
    "vlan_id": [
        1,
        30
    ],
    "_links": {
        "self": {
            "href": "/api/network/ethernet/switch/ports/RTP-SS01-510R10%28FOC22170DFR%29/Ethernet1%2F10/436212224"
        }
    }
},
"num_records": 138,
"_links": {
    "self": {
        "href": "/api/network/ethernet/switch/ports?fields=&return_records=true"
    }
}
}

```

Retrieving a ports on an ethernet switch

```

# The API:
GET /network/ethernet/switch/ports/{switch.name}/{identity.name}/
{identity.index}

# The call:
curl -X GET "https://<mgmt-ip>/api/network/ethernet/switch/ports/RTP-SS02-

```

```

510R10%28FOC22131U6T%29/Ethernet1%2F9/436211712?return_records=true" -H
"accept: application/json" -H "Content-Type: application/hal+json"

# The response:
{
  "switch": {
    "name": "RTP-SS02-510R10 (FOC22131U6T)",
    "_links": {
      "self": {
        "href": "/api/network/ethernet/switches/RTP-SS02-510R10 (FOC22131U6T)"
      }
    }
  },
  "identity": {
    "name": "Ethernet1/9",
    "index": 436211712,
    "number": 9,
    "type": "ethernetcsmacd"
  },
  "mtu": 9216,
  "duplex_type": "full_duplex",
  "speed": 100000,
  "configured": "up",
  "state": "up",
  "isl": false,
  "statistics": {
    "receive_raw": {
      "packets": 4012559315,
      "errors": 0,
      "discards": 0
    },
    "transmit_raw": {
      "packets": 337898026,
      "errors": 0,
      "discards": 0
    }
  },
  "remote_port": {
    "mtu": 9000,
    "name": "e0b",
    "device": {
      "shelf": {
        "name": "SHFFG1828000004:A",
        "uid": "12439000444923584512",
        "_links": {
          "self": {
            "href": "/api/network/ethernet/ports/e0b"
          }
        }
      }
    }
  }
}

```

```

        "self": {
            "href": "/api/storage/shelves/12439000444923584512"
        }
    }
}
},
"mac_address": "00fcbaea7228",
"vlan_id": [
    1,
    30
],
"_links": {
    "self": {
        "href": "/api/network/ethernet/switch/ports/RTP-SS02-
510R10%28FOC22131U6T%29/Ethernet1%2F9/436211712"
    }
}
}

```

Retrieve Ethernet switch ports

GET /network/ethernet/switch/ports

Introduced In: 9.8

Retrieves the ethernet switch ports.

Related ONTAP commands

- system switch ethernet interface show

Learn more

- [DOC /network/ethernet/switch/ports](#)

Parameters

| Name | Type | In | Required | Description |
|---------------|--------|-------|----------|-------------------------|
| state | string | query | False | Filter by state |
| duplex_type | string | query | False | Filter by duplex_type |
| identity.name | string | query | False | Filter by identity.name |

| Name | Type | In | Required | Description |
|-----------------------------------|---------|-------|----------|---|
| identity.index | integer | query | False | Filter by identity.index |
| identity.number | integer | query | False | Filter by identity.number |
| switch.name | string | query | False | Filter by switch.name |
| statistics.receive_ra_w.packets | integer | query | False | Filter by statistics.receive_ra_w.packets |
| statistics.receive_ra_w.discards | integer | query | False | Filter by statistics.receive_ra_w.discards |
| statistics.receive_ra_w.errors | integer | query | False | Filter by statistics.receive_ra_w.errors |
| statistics.transmit_ra_w.packets | integer | query | False | Filter by statistics.transmit_ra_w.packets |
| statistics.transmit_ra_w.discards | integer | query | False | Filter by statistics.transmit_ra_w.discards |
| statistics.transmit_ra_w.errors | integer | query | False | Filter by statistics.transmit_ra_w.errors |
| configured | string | query | False | Filter by configured |
| mtu | integer | query | False | Filter by mtu |
| isl | boolean | query | False | Filter by isl |
| vlan_id | integer | query | False | Filter by vlan_id |
| mac_address | string | query | False | Filter by mac_address |
| type | string | query | False | Filter by type |

| Name | Type | In | Required | Description |
|----------------------------------|---------------|-------|----------|---|
| speed | integer | query | False | Filter by speed |
| remote_port.mtu | integer | query | False | Filter by remote_port.mtu |
| remote_port.name | string | query | False | Filter by remote_port.name |
| remote_port.device.s helf.uid | string | query | False | Filter by remote_port.device. shelf.uid |
| remote_port.device. node.name | string | query | False | Filter by remote_port.device. node.name |
| remote_port.device. node.uuid | string | query | False | Filter by remote_port.device. node.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 | <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 |

| Name | Type | In | Required | Description |
|----------------|---------------|-------|----------|--|
| return_timeout | integer | query | False | <p>The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached.</p> <ul style="list-style-type: none"> • Default value: 1 • Max value: 120 • Min value: 0 |
| order_by | array[string] | query | False | Order results by specified fields and optional [asc] |

Response

Status: 200, Ok

| Name | Type | Description |
|-------------|--------------------|-------------------|
| _links | collection_links | |
| num_records | integer | Number of Records |
| records | array[switch_port] | |

Example response

```
{  
  "_links": {  
    "next": {  
      "href": "/api/resourcelink"  
    },  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "records": [  
    {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "configured": "string",  
      "duplex_type": "string",  
      "identity": {  
        "index": 0,  
        "name": "string",  
        "number": 0  
      },  
      "mac_address": "string",  
      "mtu": 0,  
      "remote_port": {  
        "device": {  
          "node": {  
            "_links": {  
              "self": {  
                "href": "/api/resourcelink"  
              }  
            },  
            "name": "node1",  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
          },  
          "shelf": {  
            "_links": {  
              "self": {  
                "href": "/api/resourcelink"  
              }  
            },  
            "uid": "7777841915827391056"  
          }  
        }  
      }  
    }  
  ]  
}
```

```

        },
        "mtu": 0,
        "name": "string"
    },
    "speed": 0,
    "state": "string",
    "statistics": {
        "receive_raw": {
            "discards": "100",
            "errors": "200",
            "packets": "500"
        },
        "transmit_raw": {
            "discards": "100",
            "errors": "200",
            "packets": "500"
        }
    },
    "switch": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "name": "RTP-SS01-510R03 (FOC223443KQ)"
    },
    "type": "string",
    "vlan_id": [
        "integer"
    ]
}
]
}
}

```

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 | |

collection_links

| Name | Type | Description |
|------|------|-------------|
| next | href | |
| self | href | |

self_link

| Name | Type | Description |
|------|------|-------------|
| self | href | |

identity

| Name | Type | Description |
|--------|---------|-------------------|
| index | integer | Interface Index. |
| name | string | Interface Name. |
| number | integer | Interface Number. |

_links

| Name | Type | Description |
|------|------|-------------|
| self | href | |

node

| Name | Type | Description |
|--------|--------|-------------|
| _links | _links | |
| name | string | |
| uuid | string | |

shelf

Shelf connected to this port.

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| uid | string | |

device

Device connected to port.

| Name | Type | Description |
|-------|-----------------------|-------------------------------|
| node | node | |
| shelf | shelf | Shelf connected to this port. |

remote_port

Remote port

| Name | Type | Description |
|--------|------------------------|---------------------------|
| device | device | Device connected to port. |
| mtu | integer | MTU in octets |
| name | string | Port Name. |

receive_raw

Packet receive counters for the Ethernet port.

| Name | Type | Description |
|----------|---------|------------------------------------|
| discards | integer | Total number of discarded packets. |
| errors | integer | Number of packet errors. |
| packets | integer | Total packet count. |

transmit_raw

Packet transmit counters for the Ethernet port.

| Name | Type | Description |
|----------|---------|------------------------------------|
| discards | integer | Total number of discarded packets. |

| Name | Type | Description |
|---------|---------|--------------------------|
| errors | integer | Number of packet errors. |
| packets | integer | Total packet count. |

statistics

These are raw counters for the device associated with the Ethernet port.

| Name | Type | Description |
|--------------|------------------------------|---|
| receive_raw | receive_raw | Packet receive counters for the Ethernet port. |
| transmit_raw | transmit_raw | Packet transmit counters for the Ethernet port. |

switch

The name of the specified cluster or storage switch.

| Name | Type | Description |
|--------|---------------------------|-------------|
| _links | self_link | |
| name | string | |

switch_port

Ethernet Switch Port REST API

| Name | Type | Description |
|-------------|-----------------------------|-------------------------------|
| _links | self_link | |
| configured | string | Administrative Status. |
| duplex_type | string | Duplex Settings. |
| identity | identity | |
| isl | boolean | Is configured as an ISL link. |
| mac_address | string | MAC Address. |
| mtu | integer | MTU. |
| remote_port | remote_port | Remote port |

| Name | Type | Description |
|------------|----------------|--|
| speed | integer | Interface Speed(Mbps) |
| state | string | Operational Status. |
| statistics | statistics | These are raw counters for the device associated with the Ethernet port. |
| switch | switch | The name of the specified cluster or storage switch. |
| type | string | Interface Type. |
| vlan_id | array[integer] | |

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 Ethernet switch port

GET /network/ethernet/switch/ports/{switch}/{identity.name}/{identity.index}

Introduced In: 9.8

Retrieves an ethernet switch port.

Related ONTAP commands

- system switch ethernet interface show

Parameters

| Name | Type | In | Required | Description |
|----------------|---------------|-------|----------|-------------------------------|
| switch | string | path | True | Switch Name |
| identity.name | string | path | True | Interface Name |
| identity.index | integer | path | True | Interface Index |
| fields | array[string] | query | False | Specify the fields to return. |

Response

```
Status: 200, Ok
```

| Name | Type | Description |
|-------------|-----------------------------|--|
| _links | self_link | |
| configured | string | Administrative Status. |
| duplex_type | string | Duplex Settings. |
| identity | identity | |
| isl | boolean | Is configured as an ISL link. |
| mac_address | string | MAC Address. |
| mtu | integer | MTU. |
| remote_port | remote_port | Remote port |
| speed | integer | Interface Speed(Mbps) |
| state | string | Operational Status. |
| statistics | statistics | These are raw counters for the device associated with the Ethernet port. |

| Name | Type | Description |
|---------|----------------|--|
| switch | switch | The name of the specified cluster or storage switch. |
| type | string | Interface Type. |
| vlan_id | array[integer] | |

Example response

```
{  
  "_links": {  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "configured": "string",  
  "duplex_type": "string",  
  "identity": {  
    "index": 0,  
    "name": "string",  
    "number": 0  
  },  
  "mac_address": "string",  
  "mtu": 0,  
  "remote_port": {  
    "device": {  
      "node": {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        },  
        "name": "node1",  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
      },  
      "shelf": {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        },  
        "uid": "7777841915827391056"  
      }  
    },  
    "mtu": 0,  
    "name": "string"  
  },  
  "speed": 0,  
  "state": "string",  
  "statistics": {  
    "receive_raw": {  
      "discards": "100",  
      "errors": "200",  
      "rx": "100",  
      "tx": "200"  
    }  
  }  
}
```

```

    "packets": "500"
  },
  "transmit_raw": {
    "discards": "100",
    "errors": "200",
    "packets": "500"
  }
},
"switch": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "RTP-SS01-510R03 (FOC223443KQ)"
},
"type": "string",
"vlan_id": [
  "integer"
]
}
}

```

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 | |

self_link

| Name | Type | Description |
|------|------|-------------|
| self | href | |

identity

| Name | Type | Description |
|--------|---------|-------------------|
| index | integer | Interface Index. |
| name | string | Interface Name. |
| number | integer | Interface Number. |

_links

| Name | Type | Description |
|------|------|-------------|
| self | href | |

node

| Name | Type | Description |
|--------|--------|-------------|
| _links | _links | |
| name | string | |
| uuid | string | |

shelf

Shelf connected to this port.

| Name | Type | Description |
|--------|--------|-------------|
| _links | _links | |
| uid | string | |

device

Device connected to port.

| Name | Type | Description |
|-------|-------|-------------------------------|
| node | node | |
| shelf | shelf | Shelf connected to this port. |

remote_port

Remote port

| Name | Type | Description |
|--------|---------|---------------------------|
| device | device | Device connected to port. |
| mtu | integer | MTU in octets |
| name | string | Port Name. |

receive_raw

Packet receive counters for the Ethernet port.

| Name | Type | Description |
|----------|---------|------------------------------------|
| discards | integer | Total number of discarded packets. |
| errors | integer | Number of packet errors. |
| packets | integer | Total packet count. |

transmit_raw

Packet transmit counters for the Ethernet port.

| Name | Type | Description |
|----------|---------|------------------------------------|
| discards | integer | Total number of discarded packets. |
| errors | integer | Number of packet errors. |
| packets | integer | Total packet count. |

statistics

These are raw counters for the device associated with the Ethernet port.

| Name | Type | Description |
|--------------|------------------------------|---|
| receive_raw | receive_raw | Packet receive counters for the Ethernet port. |
| transmit_raw | transmit_raw | Packet transmit counters for the Ethernet port. |

switch

The name of the specified cluster or storage switch.

| Name | Type | Description |
|--------|---------------------------|-------------|
| _links | self_link | |
| name | string | |

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 Ethernet switches

Network Ethernet switches endpoint overview

Overview

This API can be used to get information about the ethernet switches used for cluster and/or storage networks. This API support GET and PATCH calls. The GET operation returns a list of discovered switches with status and configuration information. The PATCH can be used to modify state of the switch.

Examples

Retrieving the ethernet switches for a cluster

The following example retrieves the ONTAP switches from the cluster. Note that if the `fields=*` parameter is not specified, the fields `snmp.version`, `snmp.credential`, `model`, `sw-version`, `reason`, `version-source`, `monitoring.enable`, `monitoring.subsystem.type` and `monitoring.subsystem.poll_interval` are not returned. Filters can be added on the fields to limit the results.

```
# The API:  
GET network/ethernet/switches  
  
# The call:  
curl -X GET "https://<mgmt-  
ip>/api/network/ethernet/switches?return_records=true" -H "accept:  
application/json" -H "Content-Type: application/hal+json"  
  
# The response:  
{  
  "records": [  
    {  
      "name": "RTP-CS01-510R11(FOC22092K12)",  
      "address": "172.26.207.77",  
      "discovered": true,  
      "model": "NX3232C",  
      "monitoring": {  
        "monitored": true,  
        "reason": "None"  
      },  
      "network": "cluster",  
      "serial_number": "Unknown",  
      "snmp": {  
        "version": "snmpv2c",  
        "user": "cshml!"  
      },  
      "version": "Cisco Nexus Operating System (NX-OS) Software, Version  
9.2(3)",  
      "_links": {  
        "self": {  
          "href": "/api/network/ethernet/switches/RTP-CS01-  
510R11%28FOC22092K12%29"  
        }  
      }  
    },  
    {  
      "name": "RTP-CS01-510R12(FOC22373C3P)",  
      "address": "172.26.207.82",  
      "discovered": true,  
      "model": "NX3232C",  
      "monitoring": {  
        "monitored": true,  
        "reason": "None"  
      },  
      "network": "cluster",  
      "serial_number": "Unknown",  
      "snmp": {  
        "version": "snmpv2c",  
        "user": "cshml!"  
      },  
      "version": "Cisco Nexus Operating System (NX-OS) Software, Version  
9.2(3)",  
      "_links": {  
        "self": {  
          "href": "/api/network/ethernet/switches/RTP-CS01-  
510R12%28FOC22373C3P%29"  
        }  
      }  
    }  
  ]  
}
```

```

  "discovered": true,
  "model": "NX3232C",
  "monitoring": {
    "monitored": true,
    "reason": "None"
  },
  "network": "cluster",
  "serial_number": "FOC22373C3P",
  "snmp": {
    "version": "snmpv2c",
    "user": "cshml!"
  },
  "version": "Cisco Nexus Operating System (NX-OS) Software, Version
9.2(3)",
  "_links": {
    "self": {
      "href": "/api/network/ethernet/switches/RTP-CS01-
510R12%28FOC22373C3P%29"
    }
  }
},
{
  "name": "RTP-SS01-510R10 (FOC22170DFR)",
  "address": "172.26.207.65",
  "discovered": true,
  "model": "NX3232C",
  "monitoring": {
    "monitored": true,
    "reason": "None"
  },
  "network": "storage",
  "serial_number": "FOC22170DFR",
  "snmp": {
    "version": "snmpv2c",
    "user": "cshml!"
  },
  "version": "Cisco Nexus Operating System (NX-OS) Software, Version
9.3(3)",
  "_links": {
    "self": {
      "href": "/api/network/ethernet/switches/RTP-SS01-
510R10%28FOC22170DFR%29"
    }
  }
},
{
  "name": "RTP-SS02-510R10 (FOC22131U6T)",

```

```

"address": "172.26.207.66",
"discovered": true,
"model": "NX3232C",
"monitoring": {
    "monitored": true,
    "reason": "None"
},
"network": "storage",
"serial_number": "FOC22131U6T",
"snmp": {
    "version": "snmpv2c",
    "user": "cshml!"
},
"version": "Cisco Nexus Operating System (NX-OS) Software, Version
9.3(3)",
"_links": {
    "self": {
        "href": "/api/network/ethernet/switches/RTP-SS02-
510R10%28FOC22131U6T%29"
    }
}
},
"num_records": 4,
"_links": {
    "self": {
        "href": "/api/network/ethernet/switches?fields=&return_records=true"
    }
}
}

```

Retrieving an ethernet switch for a cluster

The following example retrieves a single switch by the switchname using the API.

```

# The API:
GET /network/ethernet/switches{name}

# The call:
curl -X GET "https://<mgmt-ip>/api/network/ethernet/switches/RTP-SS02-510R10(FOC22131U6T?return_records=true" -H "accept: application/json" -H "Content-Type: application/hal+json"

# The response:
{
  "name": "RTP-SS02-510R10(FOC22131U6T)",
  "address": "172.26.207.66",
  "discovered": true,
  "model": "NX3232C",
  "monitoring": {
    "monitored": true,
    "reason": "None"
  },
  "network": "storage",
  "serial_number": "FOC22131U6T",
  "snmp": {
    "version": "snmpv2c",
    "user": "cshml1"
  },
  "version": "Cisco Nexus Operating System (NX-OS) Software, Version 9.3(3)",
  "_links": {
    "self": {
      "href": "/api/network/ethernet/switches/RTP-SS02-510R10(FOC22131U6T)"
    }
  }
}

```

Configuring a switch

The following example configures SNMP credential and version on a switch.

```

# The API:
PATCH /network/ethernet/switches{name}

# The call:
curl -X PATCH "https://<mgmt-ip>/api/network/ethernet/switches/sconqa-
corduroyl-03" -H "accept: application/json" -H "Content-Type:
application/hal+json" -d "{ \"snmp\": { \"version\": \"snmpv2c\", \"user
\": \"cshml!\"} }"
# The response:
{ }

```

Retrieve Ethernet switches attached to a chassis

GET /network/ethernet/switches

Introduced In: 9.8

Retrieves the ethernet switches attached to the chassis.

Related ONTAP commands

- system switch ethernet show

Learn more

- [DOC /network/ethernet/switches](#)

Parameters

| Name | Type | In | Required | Description |
|---------------|---------|-------|----------|-------------------------|
| serial_number | string | query | False | Filter by serial_number |
| address | string | query | False | Filter by address |
| discovered | boolean | query | False | Filter by discovered |
| snmp.version | string | query | False | Filter by snmp.version |
| snmp.user | string | query | False | Filter by snmp.user |

| Name | Type | In | Required | Description |
|----------------------|---------------|-------|----------|---|
| monitoring.monitored | boolean | query | False | Filter by monitoring.monitored |
| monitoring.reason | string | query | False | Filter by monitoring.reason |
| monitoring.enabled | boolean | query | False | Filter by monitoring.enabled |
| network | string | query | False | Filter by network |
| model | string | query | False | Filter by model |
| name | string | query | False | Filter by name |
| version | string | query | False | Filter by version |
| fields | array[string] | query | False | Specify the fields to return. |
| max_records | integer | query | False | Limit the number of records returned. |
| return_records | boolean | query | False | <p>The default is true for GET calls. When set to false, only the number of records is returned.</p> <ul style="list-style-type: none"> • Default value: 1 |

| Name | Type | In | Required | Description |
|----------------|---------------|-------|----------|--|
| return_timeout | integer | query | False | <p>The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached.</p> <ul style="list-style-type: none"> • Default value: 1 • Max value: 120 • Min value: 0 |
| order_by | array[string] | query | False | Order results by specified fields and optional [asc] |

Response

Status: 200, Ok

| Name | Type | Description |
|-------------|------------------|-------------------|
| _links | collection_links | |
| num_records | integer | Number of Records |
| records | array[switch] | |

Example response

```
{  
  "_links": {  
    "next": {  
      "href": "/api/resourcelink"  
    },  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "records": [  
    {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "address": "string",  
      "model": "string",  
      "monitoring": {  
        "reason": "string"  
      },  
      "name": "string",  
      "network": "string",  
      "serial_number": "string",  
      "snmp": {  
        "user": "string",  
        "version": "string"  
      },  
      "version": "string"  
    }  
  ]  
}
```

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 | |

collection_links

| Name | Type | Description |
|------|------|-------------|
| next | href | |
| self | href | |

self_link

| Name | Type | Description |
|------|------|-------------|
| self | href | |

monitoring

| Name | Type | Description |
|-----------|---------|----------------------------|
| enabled | boolean | Enable Health Monitoring. |
| monitored | boolean | Is Monitored. |
| reason | string | Reason For Not Monitoring. |

snmp

| Name | Type | Description |
|---------|--------|--------------------------------------|
| user | string | Community String or SNMPv3 Username. |
| version | string | SNMP Version. |

switch

Ethernet Switch REST API

| Name | Type | Description |
|---------|-----------|-------------|
| _links | self_link | |
| address | string | IP Address. |

| Name | Type | Description |
|---------------|------------|---|
| discovered | boolean | Discovered By ONTAP CDP/LLDP <ul style="list-style-type: none"> • readOnly: 1 • Introduced in: 9.8 |
| model | string | Model Number. |
| monitoring | monitoring | |
| name | string | Name. |
| network | string | Switch Network. |
| serial_number | string | Serial Number. |
| snmp | snmp | |
| version | string | Software Version. |

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 Ethernet switch details

GET /network/ethernet/switches/{name}

Introduced In: 9.8

Retrieves the details of an ethernet switch.

Related ONTAP commands

- `system switch ethernet show`

Learn more

- [DOC /network/ethernet/switches](#)

Parameters

| Name | Type | In | Required | Description |
|--------|---------------|-------|----------|-------------------------------|
| name | string | path | True | Name |
| fields | array[string] | query | False | Specify the fields to return. |

Response

```
Status: 200, Ok
```

| Name | Type | Description |
|---------------|----------------------------|---|
| _links | self_link | |
| address | string | IP Address. |
| discovered | boolean | Discovered By ONTAP CDP/LLDP <ul style="list-style-type: none">• readOnly: 1• Introduced in: 9.8 |
| model | string | Model Number. |
| monitoring | monitoring | |
| name | string | Name. |
| network | string | Switch Network. |
| serial_number | string | Serial Number. |
| snmp | snmp | |
| version | string | Software Version. |

Example response

```
{  
  "_links": {  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "address": "string",  
  "model": "string",  
  "monitoring": {  
    "reason": "string"  
  },  
  "name": "string",  
  "network": "string",  
  "serial_number": "string",  
  "snmp": {  
    "user": "string",  
    "version": "string"  
  },  
  "version": "string"  
}
```

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 | |

self_link

| Name | Type | Description |
|------|------|-------------|
| self | href | |

monitoring

| Name | Type | Description |
|-----------|---------|----------------------------|
| enabled | boolean | Enable Health Monitoring. |
| monitored | boolean | Is Monitored. |
| reason | string | Reason For Not Monitoring. |

snmp

| Name | Type | Description |
|---------|--------|--------------------------------------|
| user | string | Community String or SNMPv3 Username. |
| version | string | SNMP Version. |

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 Ethernet switch

PATCH /network/ethernet/switches/{name}

Introduced In: 9.8

Update Ethernet Switch REST API

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|-------------|
| name | string | path | True | Switch Name |

| Name | Type | In | Required | Description |
|----------------|---------|-------|----------|--|
| return_timeout | integer | query | False | <p>The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202.</p> <ul style="list-style-type: none"> • Default value: 1 • Max value: 120 • Min value: 0 |

Request Body

| Name | Type | Description |
|------------|------------|--|
| address | string | IP Address. |
| discovered | boolean | Discovered By ONTAP CDP/LLDP <ul style="list-style-type: none"> • readOnly: 1 • Introduced in: 9.8 |
| model | string | Model Number. |
| monitoring | monitoring | |

| Name | Type | Description |
|---------------|--------|-------------------|
| name | string | Name. |
| network | string | Switch Network. |
| serial_number | string | Serial Number. |
| snmp | snmp | |
| version | string | Software Version. |

Example request

```
{
  "address": "string",
  "model": "string",
  "monitoring": {
    "reason": "string"
  },
  "name": "string",
  "network": "string",
  "serial_number": "string",
  "snmp": {
    "user": "string",
    "version": "string"
  },
  "version": "string"
}
```

Response

Status: 202, Accepted

| Name | Type | Description |
|------|----------|-------------|
| job | job_link | |

Example response

```
{  
  "job": {  
    "uuid": "string"  
  }  
}
```

Error

```
Status: Default, Error
```

Definitions

See Definitions

href

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

self_link

monitoring

| Name | Type | Description |
|-----------|---------|----------------------------|
| enabled | boolean | Enable Health Monitoring. |
| monitored | boolean | Is Monitored. |
| reason | string | Reason For Not Monitoring. |

snmp

| Name | Type | Description |
|---------|--------|--------------------------------------|
| user | string | Community String or SNMPv3 Username. |
| version | string | SNMP Version. |

switch

Ethernet Switch REST API

| Name | Type | Description |
|------------|----------------------------|---|
| address | string | IP Address. |
| discovered | boolean | Discovered By ONTAP CDP/LLDP <ul style="list-style-type: none">• readOnly: 1• Introduced in: 9.8 |
| model | string | Model Number. |
| monitoring | monitoring | |
| name | string | Name. |

| Name | Type | Description |
|---------------|--------|-------------------|
| network | string | Switch Network. |
| serial_number | string | Serial Number. |
| snmp | snmp | |
| version | string | Software Version. |

_links

job_link

| Name | Type | Description |
|------|--------|---|
| uuid | string | The UUID of the asynchronous job that is triggered by a POST, PATCH, or DELETE operation. |

error_arguments

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

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 FC network interfaces

Network FC interfaces endpoint overview

Overview

Fibre Channel (FC) interfaces are the logical endpoints for FC network connections to an SVM. An FC interface provides FC access to storage within the interface SVM using either Fibre Channel Protocol or NVMe over FC (NVMe/FC).

The Fibre Channel interface REST API allows you to create, delete, update, and discover FC interfaces, and obtain status information for FC interfaces.

An FC interface is created on an FC port which is located on a cluster node. The FC port must be specified to identify the location of the interface for a POST or PATCH request that relocates an interface. You can identify the port by supplying either the node and port names or the port UUID.

Performance monitoring

Performance of an FC interface can be monitored by observing the `metric.*` and `statistics.*` properties. These properties show the performance of an FC interface 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 FC interface using the port node and name to identify the location

This example uses the `return_records` query parameter to retrieve the newly created FC interface in the POST response.

```
# The API:  
POST /api/network/fc/interfaces  
  
# The call:  
curl -X POST 'https://<mgmt-  
ip>/api/network/fc/interfaces?return_records=true' -H 'accept:  
application/hal+json' -d '{ "svm": { "name": "svm1" }, "name": "lif1",  
"location": { "home_port": { "name": "0a", "home_node": { "name": "node1" } } }, "data_protocol": "fcp" }'  
  
# The response:  
{  
  "num_records": 1,  
  "records": [  
    {  
      "svm": {  
        "uuid": "cf300f5c-db83-11e8-bd46-005056bba0e0",  
        "name": "svm1",  
        "_links": {  
          "self": {  
            "href": "/api/svm/svms/cf300f5c-db83-11e8-bd46-005056bba0e0"  
          }  
        }  
      }  
    }  
  ]  
}
```

```

} ,
"uuid": "f6045b92-dec7-11e8-a733-005056bba0e0",
"name": "lif1",
"location": {
  "home_node": {
    "uuid": "bafe9b9f-db81-11e8-bd46-005056bba0e0",
    "name": "node1",
    "_links": {
      "self": {
        "href": "/api/cluster/nodes/bafe9b9f-db81-11e8-bd46-005056bba0e0"
      }
    }
  },
  "home_port": {
    "uuid": "300c1ae3-db82-11e8-bd46-005056bba0e0",
    "name": "0a",
    "node": {
      "name": "node1"
    },
    "_links": {
      "self": {
        "href": "/api/network/fc/ports/300c1ae3-db82-11e8-bd46-005056bba0e0"
      }
    }
  },
  "node": {
    "uuid": "bafe9b9f-db81-11e8-bd46-005056bba0e0",
    "name": "node1",
    "_links": {
      "self": {
        "href": "/api/cluster/nodes/bafe9b9f-db81-11e8-bd46-005056bba0e0"
      }
    }
  },
  "port": {
    "uuid": "300c1ae3-db82-11e8-bd46-005056bba0e0",
    "name": "0a",
    "node": {
      "name": "node1"
    },
    "_links": {
      "self": {
        "href": "/api/network/fc/ports/300c1ae3-db82-11e8-bd46-005056bba0e0"
      }
    }
  }
}

```

```

005056bba0e0"
        }
    }
}
,
"enabled": true,
"state": "down",
"data_protocol": "fcp",
"wwpn": "20:04:00:50:56:bb:a0:e0",
"wwnn": "20:00:00:50:56:bb:a0:e0",
"port_address": "9da2cb1",
"_links": {
    "self": {
        "href": "/api/network/fc/interfaces/f6045b92-dec7-11e8-a733-
005056bba0e0"
    }
}
]
}

```

Creating an FC interface using the port UUID to identify the location

This example uses the `return_records` query parameter to retrieve the newly created FC interface in the POST response.

```

# The API:
POST /api/network/fc/interfaces

# The call:
curl -X POST 'https://<mgmt-
ip>/api/network/fc/interfaces?return_records=true' -H 'accept:
application/hal+json' -d '{ "svm": { "name": "svm3" }, "name": "lif2",
"location": { "home_port": { "uuid": "24bb636a-db83-11e8-9a49-
005056bb1ec6" } }, "data_protocol": "fc_nvme" }'

# The response:
{
"num_records": 1,
"records": [
{
    "svm": {
        "uuid": "a5060466-dbab-11e8-bd46-005056bba0e0",
        "name": "svm3",

```

```

"_links": {
  "self": {
    "href": "/api/svm/svms/a5060466-dbab-11e8-bd46-005056bba0e0"
  }
},
"uuid": "cdeb5591-dec9-11e8-a733-005056bba0e0",
"name": "lif2",
"location": {
  "home_node": {
    "uuid": "e85aa147-db83-11e8-9a48-005056bb1ec6",
    "name": "node3",
    "_links": {
      "self": {
        "href": "/api/cluster/nodes/e85aa147-db83-11e8-9a48-005056bb1ec6"
      }
    }
  },
  "home_port": {
    "uuid": "24bb636a-db83-11e8-9a49-005056bb1ec6",
    "name": "1b",
    "node": {
      "name": "node3"
    },
    "_links": {
      "self": {
        "href": "/api/network/fc/ports/24bb636a-db83-11e8-9a49-005056bb1ec6"
      }
    }
  },
  "node": {
    "uuid": "e85aa147-db83-11e8-9a48-005056bb1ec6",
    "name": "node3",
    "_links": {
      "self": {
        "href": "/api/cluster/nodes/e85aa147-db83-11e8-9a48-005056bb1ec6"
      }
    }
  },
  "port": {
    "uuid": "24bb636a-db83-11e8-9a49-005056bb1ec6",
    "name": "1b",
    "node": {
      "name": "node3"
    }
  }
}

```

```

        "name": "node3"
    },
    "_links": {
        "self": {
            "href": "/api/network/fc/ports/24bb636a-db83-11e8-9a49-
005056bb1ec6"
        }
    }
},
"enabled": true,
"state": "down",
"data_protocol": "fc_nvme",
"wwpn": "20:05:00:50:56:bb:a0:e0",
"wwnn": "20:02:00:50:56:bb:a0:e0",
"port_address": "612e202b",
"_links": {
    "self": {
        "href": "/api/network/fc/interfaces/cdeb5591-dec9-11e8-a733-
005056bba0e0"
    }
}
}
]
}

```

Retrieving all properties for all FC interfaces

This example uses the `fields` query parameter to retrieve all properties.

```

# The API:
GET /api/network/fc/interfaces

# The call:
curl -X GET 'https://<mgmt-ip>/api/network/fc/interfaces?fields=' -H
'accept: application/hal+json'

# The response:
{
"records": [
{
    "svm": {
        "uuid": "a5060466-dbab-11e8-bd46-005056bba0e0",
        "name": "svm3",

```

```

"_links": {
  "self": {
    "href": "/api/svm/svms/a5060466-dbab-11e8-bd46-005056bba0e0"
  }
},
"uuid": "cdeb5591-dec9-11e8-a733-005056bba0e0",
"name": "lif2",
"location": {
  "home_node": {
    "uuid": "e85aa147-db83-11e8-9a48-005056bb1ec6",
    "name": "node3",
    "_links": {
      "self": {
        "href": "/api/cluster/nodes/e85aa147-db83-11e8-9a48-005056bb1ec6"
      }
    }
  },
  "home_port": {
    "uuid": "24bb636a-db83-11e8-9a49-005056bb1ec6",
    "name": "1b",
    "node": {
      "name": "node3"
    },
    "_links": {
      "self": {
        "href": "/api/network/fc/ports/24bb636a-db83-11e8-9a49-005056bb1ec6"
      }
    }
  },
  "node": {
    "uuid": "e85aa147-db83-11e8-9a48-005056bb1ec6",
    "name": "node3",
    "_links": {
      "self": {
        "href": "/api/cluster/nodes/e85aa147-db83-11e8-9a48-005056bb1ec6"
      }
    }
  },
  "port": {
    "uuid": "24bb636a-db83-11e8-9a49-005056bb1ec6",
    "name": "1b",
    "node": {
      "name": "node3"
    }
  }
}

```

```

        "name": "node3"
    },
    "_links": {
        "self": {
            "href": "/api/network/fc/ports/24bb636a-db83-11e8-9a49-
005056bb1ec6"
        }
    }
},
"enabled": true,
"state": "down",
"data_protocol": "fc_nvme",
"wwpn": "20:05:00:50:56:bb:a0:e0",
"wwnn": "20:02:00:50:56:bb:a0:e0",
"port_address": "612e202b",
"_links": {
    "self": {
        "href": "/api/network/fc/interfaces/cdeb5591-dec9-11e8-a733-
005056bba0e0"
    }
}
},
{
    "svm": {
        "uuid": "cf300f5c-db83-11e8-bd46-005056bba0e0",
        "name": "svm1",
        "_links": {
            "self": {
                "href": "/api/svm/svms/cf300f5c-db83-11e8-bd46-005056bba0e0"
            }
        }
    },
    "uuid": "f6045b92-dec7-11e8-a733-005056bba0e0",
    "name": "lif1",
    "location": {
        "home_node": {
            "uuid": "bafe9b9f-db81-11e8-bd46-005056bba0e0",
            "name": "node1",
            "_links": {
                "self": {
                    "href": "/api/cluster/nodes/bafe9b9f-db81-11e8-bd46-
005056bba0e0"
                }
            }
        }
    }
}

```

```

"home_port": {
    "uuid": "300c1ae3-db82-11e8-bd46-005056bba0e0",
    "name": "0a",
    "node": {
        "name": "node1"
    },
    "_links": {
        "self": {
            "href": "/api/network/fc/ports/300c1ae3-db82-11e8-bd46-005056bba0e0"
        }
    }
},
"node": {
    "uuid": "bafe9b9f-db81-11e8-bd46-005056bba0e0",
    "name": "node1",
    "_links": {
        "self": {
            "href": "/api/cluster/nodes/bafe9b9f-db81-11e8-bd46-005056bba0e0"
        }
    }
},
"port": {
    "uuid": "300c1ae3-db82-11e8-bd46-005056bba0e0",
    "name": "0a",
    "node": {
        "name": "node1"
    },
    "_links": {
        "self": {
            "href": "/api/network/fc/ports/300c1ae3-db82-11e8-bd46-005056bba0e0"
        }
    }
},
"enabled": true,
"state": "down",
"data_protocol": "fcp",
"wwpn": "20:04:00:50:56:bb:a0:e0",
"wwnn": "20:00:00:50:56:bb:a0:e0",
"port_address": "9da2cb1",
"_links": {
    "self": {
        "href": "/api/network/fc/interfaces/f6045b92-dec7-11e8-a733-"
    }
}
}

```

```
005056bba0e0"
    }
}
}
],
"num_records": 2,
"_links": {
  "self": {
    "href": "/api/network/fc/interfaces?fields=*&status=up"
  }
}
}
```

Retrieving a list of selected FC interfaces

This example uses property query parameters to retrieve FC interfaces configured for the FC Protocol that are set to *up*.

```

# The API:
GET /api/network/fc/interfaces

# The call:
curl -X GET 'https://<mgmt-
ip>/api/network/fc/interfaces?data_protocol=fcp&state=up' -H 'accept:
application/hal+json'

# The response:
{
  "records": [
    {
      "svm": {
        "uuid": "cf300f5c-db83-11e8-bd46-005056bba0e0",
        "name": "svm1",
        "_links": {
          "self": {
            "href": "/api/svm/svms/cf300f5c-db83-11e8-bd46-005056bba0e0"
          }
        }
      },
      "uuid": "f6045b92-dec7-11e8-a733-005056bba0e0",
      "name": "lif1",
      "state": "up",
      "data_protocol": "fcp",
      "_links": {
        "self": {
          "href": "/api/network/fc/interfaces/f6045b92-dec7-11e8-a733-
005056bba0e0"
        }
      }
    }
  ],
  "num_records": 1,
  "_links": {
    "self": {
      "href": "/api/network/fc/interfaces?data_protocol=fcp&state=up"
    }
  }
}

```

Retrieving a specific FC interface

```
# The API:  
GET /api/network/fc/interfaces/{uuid}  
  
# The call:  
curl -X GET 'https://<mgmt-ip>/api/network/fc/interfaces/cdeb5591-dec9-  
11e8-a733-005056bba0e0' -H 'accept: application/hal+json'  
  
# The response:  
{  
  "svm": {  
    "uuid": "a5060466-dbab-11e8-bd46-005056bba0e0",  
    "name": "svm3",  
    "_links": {  
      "self": {  
        "href": "/api/svm/svms/a5060466-dbab-11e8-bd46-005056bba0e0"  
      }  
    }  
  },  
  "uuid": "cdeb5591-dec9-11e8-a733-005056bba0e0",  
  "name": "lif2",  
  "location": {  
    "home_node": {  
      "uuid": "e85aa147-db83-11e8-9a48-005056bb1ec6",  
      "name": "node3",  
      "_links": {  
        "self": {  
          "href": "/api/cluster/nodes/e85aa147-db83-11e8-9a48-005056bb1ec6"  
        }  
      }  
    },  
    "home_port": {  
      "uuid": "24bb636a-db83-11e8-9a49-005056bb1ec6",  
      "name": "1b",  
      "node": {  
        "name": "node3"  
      },  
      "_links": {  
        "self": {  
          "href": "/api/network/fc/ports/24bb636a-db83-11e8-9a49-  
005056bb1ec6"  
        }  
      }  
    },  
    "node": {  
      "name": "node3"  
    }  
  }  
}
```

```
"uuid": "e85aa147-db83-11e8-9a48-005056bb1ec6",
"name": "node3",
"_links": {
  "self": {
    "href": "/api/cluster/nodes/e85aa147-db83-11e8-9a48-005056bb1ec6"
  }
},
"port": {
  "uuid": "24bb636a-db83-11e8-9a49-005056bb1ec6",
  "name": "1b",
  "node": {
    "name": "node3"
  },
  "_links": {
    "self": {
      "href": "/api/network/fc/ports/24bb636a-db83-11e8-9a49-005056bb1ec6"
    }
  }
},
"enabled": true,
"state": "down",
"data_protocol": "fc_nvme",
"wwpn": "20:05:00:50:56:bb:a0:e0",
"wwnn": "20:02:00:50:56:bb:a0:e0",
"port_address": "612e202b",
"metric": {
  "timestamp": "2019-04-09T05:50:15Z",
  "duration": "PT15S",
  "status": "ok",
  "latency": {
    "other": 0,
    "total": 0,
    "read": 0,
    "write": 0
  },
  "iops": {
    "read": 0,
    "write": 0,
    "other": 0,
    "total": 0
  },
  "throughput": {
    "read": 0,
    "write": 0
  }
}
```

```

        "write": 0,
        "total": 0
    },
},
"statistics": {
    "timestamp": "2019-04-09T05:50:42Z",
    "status": "ok",
    "latency_raw": {
        "other": 38298,
        "total": 38298,
        "read": 0,
        "write": 0
    },
    "iops_raw": {
        "read": 0,
        "write": 0,
        "other": 3,
        "total": 3
    },
    "throughput_raw": {
        "read": 0,
        "write": 0,
        "total": 0
    }
},
"_links": {
    "self": {
        "href": "/api/network/fc/interfaces/cdeb5591-dec9-11e8-a733-005056bba0e0"
    }
}
}

```

Disabling an FC interface

When updating certain properties or deleting an FC interface, the interface must first be disabled using the following:

```

# The API:
PATCH /api/network/fc/interfaces/{uuid}

# The call:
curl -X PATCH 'https://<mgmt-ip>/api/network/fc/interfaces/f6045b92-dec7-
11e8-a733-005056bba0e0' -H 'accept: application/hal+json' -d '{ "enabled": 
false }'

```

Moving an FC interface to a new node and port

To move an FC interface to another node or port, the destination FC port must be specified in a PATCH request. Either the port UUID or node and port names can be used to identify the port.

Note that only FC interfaces configured for the FC Protocol can be moved. FC interfaces configured for NVMe/FC cannot be moved. The interface must also be set to the disabled state before being moved.

```

# The API:
PATCH /api/network/fc/interfaces/{uuid}

# The call:
curl -X PATCH 'https://<mgmt-ip>/api/network/fc/interfaces/f6045b92-dec7-
11e8-a733-005056bba0e0' -H 'accept: application/hal+json' -d '{
"location": { "home_port": { "uuid": "a1dc7aa5-db83-11e8-9ef7-
005056bbbbcc" } } }'

```

Deleting an FC interface

The FC interface must be disabled before being deleted.

```

# The API:
DELETE /api/network/fc/interfaces/{uuid}

# The call:
curl -X DELETE 'https://<mgmt-ip>/api/network/fc/interfaces/f6045b92-dec7-
11e8-a733-005056bba0e0' -H 'accept: application/hal+json'

```

Retrieve FC interfaces

GET /network/fc/interfaces

Introduced In: 9.6

Retrieves FC interfaces.

Related ONTAP commands

- `network interface show`
- `vserver fcp interface show`

Learn more

- [DOC /network/fc/interfaces](#)

Parameters

| Name | Type | In | Required | Description |
|--------------------------------------|--------|-------|----------|---|
| <code>location.port.name</code> | string | query | False | Filter by <code>location.port.name</code> |
| <code>location.port.node.name</code> | string | query | False | Filter by <code>location.port.node.name</code> |
| <code>location.port.uuid</code> | string | query | False | Filter by <code>location.port.uuid</code> |
| <code>location.home_node.name</code> | string | query | False | Filter by <code>location.home_node.name</code> <ul style="list-style-type: none">• Introduced in: 9.8 |
| <code>location.home_node.uuid</code> | string | query | False | Filter by <code>location.home_node.uuid</code> <ul style="list-style-type: none">• Introduced in: 9.8 |
| <code>location.home_port.name</code> | string | query | False | Filter by <code>location.home_port.name</code> <ul style="list-style-type: none">• Introduced in: 9.8 |

| Name | Type | In | Required | Description |
|------------------------------|---------|-------|----------|---|
| location.home_port.node.name | string | query | False | Filter by location.home_port.node.name <ul style="list-style-type: none"> • Introduced in: 9.8 |
| location.home_port.uid | string | query | False | Filter by location.home_port.uuid <ul style="list-style-type: none"> • Introduced in: 9.8 |
| location.node.name | string | query | False | Filter by location.node.name |
| location.node.uuid | string | query | False | Filter by location.node.uuid |
| location.is_home | boolean | query | False | Filter by location.is_home <ul style="list-style-type: none"> • Introduced in: 9.8 |
| enabled | boolean | query | False | Filter by enabled |
| port_address | string | query | False | Filter by port_address |
| uuid | string | query | False | Filter by uuid |
| data_protocol | string | query | False | Filter by data_protocol |
| metric.latency.total | integer | query | False | Filter by metric.latency.total <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.latency.read | integer | query | False | Filter by metric.latency.read <ul style="list-style-type: none"> • Introduced in: 9.8 |

| Name | Type | In | Required | Description |
|----------------------|---------|-------|----------|---|
| metric.latency.other | integer | query | False | Filter by metric.latency.other <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.latency.write | integer | query | False | Filter by metric.latency.write <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.status | string | query | False | Filter by metric.status <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.iops.total | integer | query | False | Filter by metric.iops.total <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.iops.read | integer | query | False | Filter by metric.iops.read <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.iops.other | integer | query | False | Filter by metric.iops.other <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.iops.write | integer | query | False | Filter by metric.iops.write <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.duration | string | query | False | Filter by metric.duration <ul style="list-style-type: none"> • Introduced in: 9.8 |

| Name | Type | In | Required | Description |
|---------------------------|---------|-------|----------|--|
| metric.timestamp | string | query | False | Filter by metric.timestamp <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.throughput.write | integer | query | False | Filter by metric.throughput.write <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.throughput.read | integer | query | False | Filter by metric.throughput.read <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.throughput.total | integer | query | False | Filter by metric.throughput.total <ul style="list-style-type: none"> • Introduced in: 9.8 |
| name | string | query | False | Filter by name |
| comment | string | query | False | Filter by comment |
| statistics.iops_raw.total | integer | query | False | Filter by statistics.iops_raw.total <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.iops_raw.read | integer | query | False | Filter by statistics.iops_raw.read <ul style="list-style-type: none"> • Introduced in: 9.8 |

| Name | Type | In | Required | Description |
|------------------------------|---------|-------|----------|---|
| statistics.iops_raw.other | integer | query | False | Filter by statistics.iops_raw.other <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.iops_raw.write | integer | query | False | Filter by statistics.iops_raw.write <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.latency_raw.total | integer | query | False | Filter by statistics.latency_raw.total <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.latency_raw.read | integer | query | False | Filter by statistics.latency_raw.read <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.latency_raw.other | integer | query | False | Filter by statistics.latency_raw.other <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.latency_raw.write | integer | query | False | Filter by statistics.latency_raw.write <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.timestamp | string | query | False | Filter by statistics.timestamp <ul style="list-style-type: none"> • Introduced in: 9.8 |

| Name | Type | In | Required | Description |
|---------------------------------|---------------|-------|----------|--|
| statistics.status | string | query | False | Filter by statistics.status <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.throughput_raw.write | integer | query | False | Filter by statistics.throughput_raw.write <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.throughput_raw.read | integer | query | False | Filter by statistics.throughput_raw.read <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.throughput_raw.total | integer | query | False | Filter by statistics.throughput_raw.total <ul style="list-style-type: none"> • Introduced in: 9.8 |
| wwpn | string | query | False | Filter by wwpn |
| wwnn | string | query | False | Filter by wnnn |
| state | string | query | False | Filter by state |
| 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. |

| Name | Type | In | Required | Description |
|----------------|---------------|-------|----------|--|
| 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 |
| 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 • Max value: 120 • Min value: 0 |
| order_by | array[string] | query | False | Order results by specified fields and optional [asc] |

Response

Status: 200, Ok

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

Example response

```
{  
  "_links": {  
    "next": {  
      "href": "/api/resourcelink"  
    },  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "records": [  
    {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "comment": "string",  
      "data_protocol": "string",  
      "location": {  
        "home_node": {  
          "_links": {  
            "self": {  
              "href": "/api/resourcelink"  
            }  
          },  
          "name": "node1",  
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        },  
        "home_port": {  
          "_links": {  
            "self": {  
              "href": "/api/resourcelink"  
            }  
          },  
          "name": "0a",  
          "node": {  
            "name": "node1"  
          },  
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        },  
        "node": {  
          "_links": {  
            "self": {  
              "href": "/api/resourcelink"  
            }  
          }  
        }  
      }  
    }  
  ]  
}
```

```

        }
    },
    "name": "node1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"port": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "name": "0a",
    "node": {
        "name": "node1"
    },
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
},
"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": "2017-01-25T11:20:13Z"
},
"name": "lif1",
"port_address": "5060F",
"state": "string",

```

```

"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": "2017-01-25T11:20:13Z"
},
"svm": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "name": "svml",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
},
"uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",
"wwnn": "20:00:00:50:56:b4:13:01",
"wwpn": "20:00:00:50:56:b4:13:a8"
}
]
}

```

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 | |

home_node

| Name | Type | Description |
|--------|--------|-------------|
| _links | _links | |
| name | string | |
| uuid | string | |

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. |

fc_port_reference

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. |

| Name | Type | Description |
|------|--------|---------------------------------------|
| uuid | string | The unique identifier of the FC port. |

node

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| uuid | string | |

location

The location of the FC interface is defined by the location of its port. An FC port is identified by its UUID, or a combination of its node name and port name. Either the UUID or the node name and port name are required for POST. To move an interface, supply either the UUID or the node name and port name in a PATCH.

The location of an FC interface can be set using "location.home_node" and "location.home_port" during a POST or PATCH. "location.node" and "location.port" refer to the current location of the FC interface. This may be different from "location.home_node" and "location.home_port" in instances where the FC interface failed over due to an offline node.

| Name | Type | Description |
|-----------|-----------------------------------|--|
| home_node | home_node | |
| home_port | fc_port_reference | An FC port is the physical port of an FC adapter on a cluster node that can be connected to an FC network. |
| is_home | boolean | Indicates whether or not the FC interface currently resides on the home node. |
| node | node | |
| port | fc_port_reference | An FC port is the physical port of an FC adapter on a cluster node that can be connected to an FC network. |

iops

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

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

latency

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

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

throughput

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

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

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

metric

Performance numbers, such as IOPS latency and throughput

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

| Name | Type | Description |
|------------|------------|---|
| throughput | throughput | The rate of throughput bytes per second observed at the storage object. |
| timestamp | string | The timestamp of the performance data. |

iops_raw

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

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

latency_raw

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

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

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

throughput_raw

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

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

statistics

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

| Name | Type | Description |
|-------------|-------------|--|
| iops_raw | 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. |

| Name | Type | Description |
|----------------|--------------------------------|--|
| 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. |

fc_interface

A Fibre Channel (FC) interface is the logical endpoint for FC network connections to an SVM. An FC interface provides FC access to storage within the interface SVM using either Fibre Channel Protocol or NVMe over Fibre Channel (NVMe/FC).

An FC interface is created on an FC port which is located on a cluster node. The FC port must be specified to identify the location of the interface for a POST or PATCH operation that relocates an interface. You can identify the port by supplying either the node and port names or the port UUID.

| Name | Type | Description |
|---------------|--------------------------|--|
| _links | _links | |
| comment | string | A user configurable comment. Optional in POST; valid in PATCH. To clear a prior comment, set the property to an empty string in PATCH. |
| data_protocol | string | The data protocol for which the FC interface is configured. Required in POST. |
| enabled | boolean | The administrative state of the FC interface. The FC interface can be disabled to block all FC communication with the SVM through this interface. Optional in POST and PATCH; defaults to <i>true</i> (enabled) in POST. |
| location | location | <p>The location of the FC interface is defined by the location of its port. An FC port is identified by its UUID, or a combination of its node name and port name. Either the UUID or the node name and port name are required for POST. To move an interface, supply either the UUID or the node name and port name in a PATCH.</p> <p>The location of an FC interface can be set using "location.home_node" and "location.home_port" during a POST or PATCH. "location.node" and "location.port" refer to the current location of the FC interface. This may be different from "location.home_node" and "location.home_port" in instances where the FC interface failed over due to an offline node.</p> |
| metric | metric | Performance numbers, such as IOPS latency and throughput |

| Name | Type | Description |
|--------------|------------|---|
| name | string | The name of the FC interface. Required in POST; optional in PATCH. |
| port_address | string | <p>The port address of the FC interface. Each FC port in an FC switched fabric has its own unique FC port address for routing purposes. The FC port address is assigned by a switch in the fabric when that port logs in to the fabric. This property refers to the address given by a switch to the FC interface when the SVM performs a port login (PLOGI).</p> <p>This is useful for obtaining statistics and diagnostic information from FC switches.</p> <p>This is a hexadecimal encoded numeric value.</p> |
| state | string | <p>The current operational state of the FC interface. The state is set to <i>down</i> if the interface is not enabled.</p> <p>If the node hosting the port is down or unavailable, no state value is returned.</p> |
| statistics | statistics | These are raw performance numbers, such as IOPS latency and throughput. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster. |
| svm | svm | |
| uuid | string | The unique identifier of the FC interface. Required in the URL. |

| Name | Type | Description |
|------|--------|---|
| wwnn | string | <p>The world wide node name (WWNN) of the FC interface SVM. The WWNN is generated by ONTAP when Fibre Channel Protocol or the NVMe service is created for the FC interface SVM.</p> <ul style="list-style-type: none"> • example: 20:00:00:50:56:b4:13:01 • readOnly: 1 • Introduced in: 9.6 |
| wwpn | string | <p>The world wide port name (WWPN) of the FC interface. The WWPN is generated by ONTAP when the FC interface is created.</p> <ul style="list-style-type: none"> • example: 20:00:00:50:56:b4:13:a8 • readOnly: 1 • Introduced in: 9.6 |

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 FC interface

POST /network/fc/interfaces

Introduced In: 9.6

Creates an FC interface.

Required properties

- `svm.uuid` or `svm.name` - Existing SVM in which to create the FC interface.
- `name` - Name of the FC interface.
- `location.port.uuid` or both `location.port.name` and `location.port.node.name` - FC port on which to create the FC interface.
- `data_protocol` - Data protocol for the FC interface.

Default property values

If not specified in POST, the following default property values are assigned.

- `enabled` - `true`

Related ONTAP commands

- `network interface create`

Learn more

- [DOC /network/fc/interfaces](#)

Parameters

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

Request Body

| Name | Type | Description |
|---------|--------|--|
| comment | string | A user configurable comment. Optional in POST; valid in PATCH. To clear a prior comment, set the property to an empty string in PATCH. |

| Name | Type | Description |
|---------------|----------|--|
| data_protocol | string | The data protocol for which the FC interface is configured. Required in POST. |
| enabled | boolean | The administrative state of the FC interface. The FC interface can be disabled to block all FC communication with the SVM through this interface. Optional in POST and PATCH; defaults to <i>true</i> (enabled) in POST. |
| location | location | <p>The location of the FC interface is defined by the location of its port. An FC port is identified by its UUID, or a combination of its node name and port name. Either the UUID or the node name and port name are required for POST. To move an interface, supply either the UUID or the node name and port name in a PATCH.</p> <p>The location of an FC interface can be set using "location.home_node" and "location.home_port" during a POST or PATCH. "location.node" and "location.port" refer to the current location of the FC interface. This may be different from "location.home_node" and "location.home_port" in instances where the FC interface failed over due to an offline node.</p> |
| name | string | The name of the FC interface. Required in POST; optional in PATCH. |

| Name | Type | Description |
|--------------|---------------------|---|
| port_address | string | <p>The port address of the FC interface. Each FC port in an FC switched fabric has its own unique FC port address for routing purposes. The FC port address is assigned by a switch in the fabric when that port logs in to the fabric. This property refers to the address given by a switch to the FC interface when the SVM performs a port login (PLOGI).</p> <p>This is useful for obtaining statistics and diagnostic information from FC switches.</p> <p>This is a hexadecimal encoded numeric value.</p> |
| state | string | <p>The current operational state of the FC interface. The state is set to <i>down</i> if the interface is not enabled.</p> <p>If the node hosting the port is down or unavailable, no state value is returned.</p> |
| svm | svm | |
| uuid | string | The unique identifier of the FC interface. Required in the URL. |
| wwnn | string | <p>The world wide node name (WWNN) of the FC interface SVM. The WWNN is generated by ONTAP when Fibre Channel Protocol or the NVMe service is created for the FC interface SVM.</p> <ul style="list-style-type: none"> example: 20:00:00:50:56:b4:13:01 readOnly: 1 Introduced in: 9.6 |

| Name | Type | Description |
|------|--------|--|
| wwpn | string | <p>The world wide port name (WWPN) of the FC interface. The WWPN is generated by ONTAP when the FC interface is created.</p> <ul style="list-style-type: none"> • example: 20:00:00:50:56:b4:13:a8 • readOnly: 1 • Introduced in: 9.6 |

Example request

```
{  
    "comment": "string",  
    "data_protocol": "string",  
    "location": {  
        "home_node": {  
            "name": "node1",  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        },  
        "home_port": {  
            "name": "0a",  
            "node": {  
                "name": "node1"  
            },  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        },  
        "node": {  
            "name": "node1",  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        },  
        "port": {  
            "name": "0a",  
            "node": {  
                "name": "node1"  
            },  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        }  
    },  
    "name": "lif1",  
    "port_address": "5060F",  
    "state": "string",  
    "svm": {  
        "name": "svml",  
        "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
    },  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",  
    "wwnn": "20:00:00:50:56:b4:13:01",  
    "wwpn": "20:00:00:50:56:b4:13:a8"  
}
```

Response

Status: 201, Created

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

Example response

```
{  
  "records": [  
    {  
      "comment": "string",  
      "data_protocol": "string",  
      "location": {  
        "home_node": {  
          "name": "node1",  
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        },  
        "home_port": {  
          "name": "0a",  
          "node": {  
            "name": "node1"  
          },  
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        },  
        "node": {  
          "name": "node1",  
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        },  
        "port": {  
          "name": "0a",  
          "node": {  
            "name": "node1"  
          },  
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        }  
      },  
      "name": "lif1",  
      "port_address": "5060F",  
      "state": "string",  
      "svm": {  
        "name": "svml1",  
        "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
      },  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",  
      "wwnn": "20:00:00:50:56:b4:13:01",  
      "wwpn": "20:00:00:50:56:b4:13:a8"  
    }  
  ]  
}
```

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|---|
| 1966140 | An interface with the same name already exists. |
| 1966217 | The specified port is not valid on the node provided. |
| 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. |
| 5373966 | A Fibre Channel interface with the <code>fcp</code> protocol cannot be created in an SVM that is configured for NVMe. |
| 5374102 | The specified Fibre Channel interface cannot be created because the Fibre Channel adapter is down. Bring the adapter up and try again. |
| 5374871 | The Fibre Channel port identified by the specified UUID does not refer to the same port as that identified by the specified node name and/or port name. |
| 5374872 | If either <code>location.port.node.name</code> or <code>location.port.name</code> is supplied, both properties must be supplied. |
| 5374873 | The Fibre Channel port must be specified using either <code>location.port.uuid</code> or <code>location.port.node.name</code> and <code>location.port.name</code> . |
| 72089652 | An NVMe service must be created before creating a Fibre Channel interface using the NVMe over FC data protocol. |
| 72089672 | The specified Fibre Channel port does not support the NVMe over FC data protocol. |
| 72089900 | A Fibre Channel interface with the <code>fc_nvme</code> protocol cannot be created in an SVM that is configured for a SAN protocol. |

Definitions

See Definitions

href

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

_links

home_node

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| uuid | string | |

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. |

fc_port_reference

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 |
|------|--------|---|
| 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. |

node

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| uuid | string | |

location

The location of the FC interface is defined by the location of its port. An FC port is identified by its UUID,

or a combination of its node name and port name. Either the UUID or the node name and port name are required for POST. To move an interface, supply either the UUID or the node name and port name in a PATCH.

The location of an FC interface can be set using "location.home_node" and "location.home_port" during a POST or PATCH. "location.node" and "location.port" refer to the current location of the FC interface. This may be different from "location.home_node" and "location.home_port" in instances where the FC interface failed over due to an offline node.

| Name | Type | Description |
|-----------|-------------------|--|
| home_node | home_node | |
| home_port | fc_port_reference | An FC port is the physical port of an FC adapter on a cluster node that can be connected to an FC network. |
| is_home | boolean | Indicates whether or not the FC interface currently resides on the home node. |
| node | node | |
| port | fc_port_reference | An FC port is the physical port of an FC adapter on a cluster node that can be connected to an FC network. |

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

Performance numbers, such as IOPS latency and throughput

| 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. |

| 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

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

| Name | Type | Description |
|----------------|--------------------------------|---|
| iops_raw | 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. |

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

svm

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

fc_interface

A Fibre Channel (FC) interface is the logical endpoint for FC network connections to an SVM. An FC interface provides FC access to storage within the interface SVM using either Fibre Channel Protocol or NVMe over Fibre Channel (NVMe/FC).

An FC interface is created on an FC port which is located on a cluster node. The FC port must be specified to identify the location of the interface for a POST or PATCH operation that relocates an interface. You can identify the port by supplying either the node and port names or the port UUID.

| Name | Type | Description |
|---------------|---------|--|
| comment | string | A user configurable comment. Optional in POST; valid in PATCH. To clear a prior comment, set the property to an empty string in PATCH. |
| data_protocol | string | The data protocol for which the FC interface is configured. Required in POST. |
| enabled | boolean | The administrative state of the FC interface. The FC interface can be disabled to block all FC communication with the SVM through this interface. Optional in POST and PATCH; defaults to <i>true</i> (enabled) in POST. |

| Name | Type | Description |
|--------------|----------|--|
| location | location | <p>The location of the FC interface is defined by the location of its port. An FC port is identified by its UUID, or a combination of its node name and port name. Either the UUID or the node name and port name are required for POST. To move an interface, supply either the UUID or the node name and port name in a PATCH.</p> <p>The location of an FC interface can be set using "location.home_node" and "location.home_port" during a POST or PATCH. "location.node" and "location.port" refer to the current location of the FC interface. This may be different from "location.home_node" and "location.home_port" in instances where the FC interface failed over due to an offline node.</p> |
| name | string | <p>The name of the FC interface. Required in POST; optional in PATCH.</p> |
| port_address | string | <p>The port address of the FC interface. Each FC port in an FC switched fabric has its own unique FC port address for routing purposes. The FC port address is assigned by a switch in the fabric when that port logs in to the fabric. This property refers to the address given by a switch to the FC interface when the SVM performs a port login (PLOGI).</p> <p>This is useful for obtaining statistics and diagnostic information from FC switches.</p> <p>This is a hexadecimal encoded numeric value.</p> |

| Name | Type | Description |
|-------|---------------------|---|
| state | string | <p>The current operational state of the FC interface. The state is set to <i>down</i> if the interface is not enabled.</p> <p>If the node hosting the port is down or unavailable, no state value is returned.</p> |
| svm | svm | |
| uuid | string | The unique identifier of the FC interface. Required in the URL. |
| wwnn | string | <p>The world wide node name (WWNN) of the FC interface SVM. The WWNN is generated by ONTAP when Fibre Channel Protocol or the NVMe service is created for the FC interface SVM.</p> <ul style="list-style-type: none"> • example: 20:00:00:50:56:b4:13:01 • readOnly: 1 • Introduced in: 9.6 |
| wwpn | string | <p>The world wide port name (WWPN) of the FC interface. The WWPN is generated by ONTAP when the FC interface is created.</p> <ul style="list-style-type: none"> • example: 20:00:00:50:56:b4:13:a8 • readOnly: 1 • Introduced in: 9.6 |

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 FC interface

DELETE /network/fc/interfaces/{uuid}

Introduced In: 9.6

Deletes an FC interface.

Related ONTAP commands

- `network interface delete`

Learn more

- [DOC /network/fc/interfaces](#)

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|---|
| uuid | string | path | True | The unique identifier for the FC interface. |

Response

Status: 200, Ok

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|--|
| 53280992 | The FC interface could not be deleted because it is enabled. |

| 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 FC interface

GET /network/fc/interfaces/{uuid}

Introduced In: 9.6

Retrieves an FC interface.

Expensive properties

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

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

Related ONTAP commands

- `network interface show`
- `vserver fcp interface show`

Learn more

- [DOC /network/fc/interfaces](#)

Parameters

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

Response

Status: 200, Ok

| Name | Type | Description |
|---------------|------------------------|--|
| _links | _links | |
| comment | string | A user configurable comment. Optional in POST; valid in PATCH. To clear a prior comment, set the property to an empty string in PATCH. |
| data_protocol | string | The data protocol for which the FC interface is configured. Required in POST. |
| enabled | boolean | The administrative state of the FC interface. The FC interface can be disabled to block all FC communication with the SVM through this interface. Optional in POST and PATCH; defaults to <i>true</i> (enabled) in POST. |

| Name | Type | Description |
|--------------|----------|--|
| location | location | <p>The location of the FC interface is defined by the location of its port. An FC port is identified by its UUID, or a combination of its node name and port name. Either the UUID or the node name and port name are required for POST. To move an interface, supply either the UUID or the node name and port name in a PATCH.</p> <p>The location of an FC interface can be set using "location.home_node" and "location.home_port" during a POST or PATCH. "location.node" and "location.port" refer to the current location of the FC interface. This may be different from "location.home_node" and "location.home_port" in instances where the FC interface failed over due to an offline node.</p> |
| metric | metric | Performance numbers, such as IOPS latency and throughput |
| name | string | The name of the FC interface. Required in POST; optional in PATCH. |
| port_address | string | <p>The port address of the FC interface. Each FC port in an FC switched fabric has its own unique FC port address for routing purposes. The FC port address is assigned by a switch in the fabric when that port logs in to the fabric. This property refers to the address given by a switch to the FC interface when the SVM performs a port login (PLOGI).</p> <p>This is useful for obtaining statistics and diagnostic information from FC switches.</p> <p>This is a hexadecimal encoded numeric value.</p> |

| Name | Type | Description |
|------------|------------|---|
| state | string | <p>The current operational state of the FC interface. The state is set to <i>down</i> if the interface is not enabled.</p> <p>If the node hosting the port is down or unavailable, no state value is returned.</p> |
| statistics | statistics | <p>These are raw performance numbers, such as IOPS latency and throughput. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.</p> |
| svm | svm | |
| uuid | string | <p>The unique identifier of the FC interface. Required in the URL.</p> |
| wwnn | string | <p>The world wide node name (WWNN) of the FC interface SVM. The WWNN is generated by ONTAP when Fibre Channel Protocol or the NVMe service is created for the FC interface SVM.</p> <ul style="list-style-type: none"> example: 20:00:00:50:56:b4:13:01 readOnly: 1 Introduced in: 9.6 |
| wwpn | string | <p>The world wide port name (WWPN) of the FC interface. The WWPN is generated by ONTAP when the FC interface is created.</p> <ul style="list-style-type: none"> example: 20:00:00:50:56:b4:13:a8 readOnly: 1 Introduced in: 9.6 |

Example response

```
{  
  "_links": {  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "comment": "string",  
  "data_protocol": "string",  
  "location": {  
    "home_node": {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "name": "node1",  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    },  
    "home_port": {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "name": "0a",  
      "node": {  
        "name": "node1"  
      },  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    },  
    "node": {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "name": "node1",  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    },  
    "port": {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      }  
    }  
  }  
}
```

```
        },
        "name": "0a",
        "node": {
            "name": "node1"
        },
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
},
"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": "2017-01-25T11:20:13Z"
},
"name": "lif1",
"port_address": "5060F",
"state": "string",
"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": "2017-01-25T11:20:13Z"
},
"svm": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "svml",
  "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
},
"uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",
"wwnn": "20:00:00:50:56:b4:13:01",
"wwpn": "20:00:00:50:56:b4:13:a8"
}
}

```

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 | |

home_node

| Name | Type | Description |
|--------|--------|-------------|
| _links | _links | |
| name | string | |
| uuid | string | |

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. |

fc_port_reference

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. |

node

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| uuid | string | |

location

The location of the FC interface is defined by the location of its port. An FC port is identified by its UUID, or a combination of its node name and port name. Either the UUID or the node name and port name are required for POST. To move an interface, supply either the UUID or the node name and port name in a PATCH.

The location of an FC interface can be set using "location.home_node" and "location.home_port" during a POST or PATCH. "location.node" and "location.port" refer to the current location of the FC interface. This may be different from "location.home_node" and "location.home_port" in instances where the FC interface failed over due to an offline node.

| Name | Type | Description |
|-----------|-----------------------------------|--|
| home_node | home_node | |
| home_port | fc_port_reference | An FC port is the physical port of an FC adapter on a cluster node that can be connected to an FC network. |
| is_home | boolean | Indicates whether or not the FC interface currently resides on the home node. |
| node | node | |
| port | fc_port_reference | An FC port is the physical port of an FC adapter on a cluster node that can be connected to an FC network. |

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. |

| 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

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

Performance numbers, such as IOPS latency and throughput

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

latency_raw

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

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

throughput_raw

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

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

statistics

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

| Name | Type | Description |
|-------------|-----------------------------|--|
| iops_raw | 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. |

| Name | Type | Description |
|----------------|--------------------------------|--|
| 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. |

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 FC interface

PATCH /network/fc/interfaces/{uuid}

Introduced In: 9.6

Updates an FC interface.

Related ONTAP commands

- `network interface modify`

Learn more

- [DOC /network/fc/interfaces](#)

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|---|
| uuid | string | path | True | The unique identifier for the FC interface. |

Request Body

| Name | Type | Description |
|----------|----------|--|
| comment | string | A user configurable comment. Optional in POST; valid in PATCH. To clear a prior comment, set the property to an empty string in PATCH. |
| enabled | boolean | The administrative state of the FC interface. The FC interface can be disabled to block all FC communication with the SVM through this interface. Optional in POST and PATCH; defaults to <i>true</i> (enabled) in POST. |
| location | location | <p>The location of the FC interface is defined by the location of its port. An FC port is identified by its UUID, or a combination of its node name and port name. Either the UUID or the node name and port name are required for POST. To move an interface, supply either the UUID or the node name and port name in a PATCH.</p> <p>The location of an FC interface can be set using "location.home_node" and "location.home_port" during a POST or PATCH. "location.node" and "location.port" refer to the current location of the FC interface. This may be different from "location.home_node" and "location.home_port" in instances where the FC interface failed over due to an offline node.</p> |
| name | string | The name of the FC interface. Required in POST; optional in PATCH. |

| Name | Type | Description |
|--------------|---------------------|---|
| port_address | string | <p>The port address of the FC interface. Each FC port in an FC switched fabric has its own unique FC port address for routing purposes. The FC port address is assigned by a switch in the fabric when that port logs in to the fabric. This property refers to the address given by a switch to the FC interface when the SVM performs a port login (PLOGI).</p> <p>This is useful for obtaining statistics and diagnostic information from FC switches.</p> <p>This is a hexadecimal encoded numeric value.</p> |
| state | string | <p>The current operational state of the FC interface. The state is set to <i>down</i> if the interface is not enabled.</p> <p>If the node hosting the port is down or unavailable, no state value is returned.</p> |
| svm | svm | |
| uuid | string | The unique identifier of the FC interface. Required in the URL. |
| wwnn | string | <p>The world wide node name (WWNN) of the FC interface SVM. The WWNN is generated by ONTAP when Fibre Channel Protocol or the NVMe service is created for the FC interface SVM.</p> <ul style="list-style-type: none"> example: 20:00:00:50:56:b4:13:01 readOnly: 1 Introduced in: 9.6 |

| Name | Type | Description |
|------|--------|--|
| wwpn | string | <p>The world wide port name (WWPN) of the FC interface. The WWPN is generated by ONTAP when the FC interface is created.</p> <ul style="list-style-type: none"> • example: 20:00:00:50:56:b4:13:a8 • readOnly: 1 • Introduced in: 9.6 |

Example request

```
{  
    "comment": "string",  
    "location": {  
        "home_node": {  
            "name": "node1",  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        },  
        "home_port": {  
            "name": "0a",  
            "node": {  
                "name": "node1"  
            },  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        },  
        "node": {  
            "name": "node1",  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        },  
        "port": {  
            "name": "0a",  
            "node": {  
                "name": "node1"  
            },  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        }  
    },  
    "name": "lif1",  
    "port_address": "5060F",  
    "state": "string",  
    "svm": {  
        "name": "svml1",  
        "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
    },  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",  
    "wwnn": "20:00:00:50:56:b4:13:01",  
    "wwpn": "20:00:00:50:56:b4:13:a8"  
}
```

Response

Status: 200, Ok

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|---|
| 1966140 | An interface with the same name already exists. |
| 1966217 | The specified port is not valid on the node provided. |
| 1966238 | The node or port of an active SAN data interface cannot be changed. |
| 1966702 | The destination node is not healthy. |
| 5374579 | The SAN Kernel Agent on the node is unavailable. |
| 5374870 | A partial failure occurred; renaming the interface failed. Correct the error and resubmit the request. |
| 5374871 | The Fibre Channel port identified by the specified UUID does not refer to the same port as that identified by the specified node name and/or port name. |
| 5374872 | If either <code>location.port.node.name</code> or <code>location.port.name</code> is supplied, both properties must be supplied. |
| 72089674 | You cannot move a Fibre Channel interface configured for the NVMe over FC data protocol. |

Definitions

See Definitions

href

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

_links

home_node

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| uuid | string | |

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. |

fc_port_reference

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 |
|------|--------|---|
| 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. |

node

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| uuid | string | |

location

The location of the FC interface is defined by the location of its port. An FC port is identified by its UUID,

or a combination of its node name and port name. Either the UUID or the node name and port name are required for POST. To move an interface, supply either the UUID or the node name and port name in a PATCH.

The location of an FC interface can be set using "location.home_node" and "location.home_port" during a POST or PATCH. "location.node" and "location.port" refer to the current location of the FC interface. This may be different from "location.home_node" and "location.home_port" in instances where the FC interface failed over due to an offline node.

| Name | Type | Description |
|-----------|-------------------|--|
| home_node | home_node | |
| home_port | fc_port_reference | An FC port is the physical port of an FC adapter on a cluster node that can be connected to an FC network. |
| is_home | boolean | Indicates whether or not the FC interface currently resides on the home node. |
| node | node | |
| port | fc_port_reference | An FC port is the physical port of an FC adapter on a cluster node that can be connected to an FC network. |

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

Performance numbers, such as IOPS latency and throughput

| 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. |

| 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

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

| Name | Type | Description |
|----------------|--------------------------------|---|
| iops_raw | 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. |

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

svm

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

fc_interface

A Fibre Channel (FC) interface is the logical endpoint for FC network connections to an SVM. An FC interface provides FC access to storage within the interface SVM using either Fibre Channel Protocol or NVMe over Fibre Channel (NVMe/FC).

An FC interface is created on an FC port which is located on a cluster node. The FC port must be specified to identify the location of the interface for a POST or PATCH operation that relocates an interface. You can identify the port by supplying either the node and port names or the port UUID.

| Name | Type | Description |
|---------|---------|--|
| comment | string | A user configurable comment. Optional in POST; valid in PATCH. To clear a prior comment, set the property to an empty string in PATCH. |
| enabled | boolean | The administrative state of the FC interface. The FC interface can be disabled to block all FC communication with the SVM through this interface. Optional in POST and PATCH; defaults to <i>true</i> (enabled) in POST. |

| Name | Type | Description |
|--------------|----------|--|
| location | location | <p>The location of the FC interface is defined by the location of its port. An FC port is identified by its UUID, or a combination of its node name and port name. Either the UUID or the node name and port name are required for POST. To move an interface, supply either the UUID or the node name and port name in a PATCH.</p> <p>The location of an FC interface can be set using "location.home_node" and "location.home_port" during a POST or PATCH. "location.node" and "location.port" refer to the current location of the FC interface. This may be different from "location.home_node" and "location.home_port" in instances where the FC interface failed over due to an offline node.</p> |
| name | string | <p>The name of the FC interface. Required in POST; optional in PATCH.</p> |
| port_address | string | <p>The port address of the FC interface. Each FC port in an FC switched fabric has its own unique FC port address for routing purposes. The FC port address is assigned by a switch in the fabric when that port logs in to the fabric. This property refers to the address given by a switch to the FC interface when the SVM performs a port login (PLOGI).</p> <p>This is useful for obtaining statistics and diagnostic information from FC switches.</p> <p>This is a hexadecimal encoded numeric value.</p> |

| Name | Type | Description |
|-------|---------------------|---|
| state | string | <p>The current operational state of the FC interface. The state is set to <i>down</i> if the interface is not enabled.</p> <p>If the node hosting the port is down or unavailable, no state value is returned.</p> |
| svm | svm | |
| uuid | string | The unique identifier of the FC interface. Required in the URL. |
| wwnn | string | <p>The world wide node name (WWNN) of the FC interface SVM. The WWNN is generated by ONTAP when Fibre Channel Protocol or the NVMe service is created for the FC interface SVM.</p> <ul style="list-style-type: none"> • example: 20:00:00:50:56:b4:13:01 • readOnly: 1 • Introduced in: 9.6 |
| wwpn | string | <p>The world wide port name (WWPN) of the FC interface. The WWPN is generated by ONTAP when the FC interface is created.</p> <ul style="list-style-type: none"> • example: 20:00:00:50:56:b4:13:a8 • readOnly: 1 • Introduced in: 9.6 |

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 FC interface historical performance metrics

GET /network/fc/interfaces/{uuid}/metrics

Introduced In: 9.8

Retrieves historical performance metrics for an FC interface.

Parameters

| Name | Type | In | Required | Description |
|------------------|---------|-------|----------|----------------------------|
| 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 |
| timestamp | string | query | False | Filter by timestamp |
| status | string | query | False | Filter by status |
| iops.total | integer | query | False | Filter by iops.total |
| iops.read | integer | query | False | Filter by iops.read |
| iops.other | integer | query | False | Filter by iops.other |
| iops.write | integer | query | False | Filter by iops.write |

| Name | Type | In | Required | Description |
|---------------|---------|-------|----------|---|
| latency.total | integer | query | False | Filter by latency.total |
| latency.read | integer | query | False | Filter by latency.read |
| latency.other | integer | query | False | Filter by latency.other |
| latency.write | integer | query | False | Filter by latency.write |
| uuid | string | path | True | Unique identifier of the FC interface. |
| interval | string | query | False | <p>The time range for the data. Examples can be 1h, 1d, 1m, 1w, 1y. The period for each time range is as follows:</p> <ul style="list-style-type: none"> • 1h: Metrics over the most recent hour sampled over 15 seconds. • 1d: Metrics over the most recent day sampled over 5 minutes. • 1w: Metrics over the most recent week sampled over 30 minutes. • 1m: Metrics over the most recent month sampled over 2 hours. • 1y: Metrics over the most recent year sampled over a day. • Default value: 1 • enum: ["1h", "1d", "1w", "1m", "1y"] |

| Name | Type | In | Required | Description |
|--|----------------|---------|----------|--|
| return_timeout | integer | query | False | <p>The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached.</p> <ul style="list-style-type: none"> • Default value: 1 • Max value: 120 • Min value: 0 |
| fields | array[string] | query | False | Specify the fields to return. |
| max_records | integer | query | False | Limit the number of records returned. |
| order_by | array[string] | query | False | Order results by specified fields and optional [asc] |
| desc] direction. Default direction is 'asc' for ascending. | return_records | boolean | query | False |

Response

Status: 200, Ok

| Name | Type | Description |
|-------------|------------------------|-------------------|
| _links | _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": "2017-01-25T11:20:13Z"  
    }  
  ]  
}
```

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. |

records

Performance numbers, such as IOPS latency and throughput

| 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. |

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 FC port information

Network FC ports endpoint overview

Overview

Fibre Channel (FC) ports are the physical ports of FC adapters on ONTAP cluster nodes that can be connected to FC networks to provide FC network connectivity. An FC port defines the location of an FC interface within the ONTAP cluster.

The Fibre Channel port REST API allows you to discover FC ports, obtain status information for FC ports, and configure FC port properties. POST and DELETE requests are not supported. You must physically add and remove FC adapters to ONTAP nodes to create and remove ports from the ONTAP cluster.

Performance monitoring

Performance of an FC port can be monitored by observing the `metric.*` and `statistics.*` properties. These properties show the performance of an FC port 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

Retrieving all FC ports

```
# The API:
GET /api/network/fc/ports

# The call:
curl -X GET "https://<mgmt-ip>/api/network/fc/ports" -H "accept:
application/hal+json"

# The response:
{
  "records": [
    {
      "name": "port1",
      "type": "FC",
      "status": "Up",
      "location": "node1"
    },
    {
      "name": "port2",
      "type": "FC",
      "status": "Up",
      "location": "node2"
    }
  ]
}
```

```

"node": {
    "name": "node1",
    "uuid": "3c768e01-1abc-4b3b-b7c0-629ceb62a497",
    "_links": {
        "self": {
            "href": "/api/cluster/nodes/3c768e01-1abc-4b3b-b7c0-629ceb62a497"
        }
    }
},
"uuid": "931b20f8-b047-11e8-9af3-005056bb838e",
"name": "0a",
"_links": {
    "self": {
        "href": "/api/network/fc/ports/931b20f8-b047-11e8-9af3-005056bb838e"
    }
}
},
{
"node": {
    "name": "node1",
    "uuid": "3c768e01-1abc-4b3b-b7c0-629ceb62a497",
    "_links": {
        "self": {
            "href": "/api/cluster/nodes/3c768e01-1abc-4b3b-b7c0-629ceb62a497"
        }
    }
},
"uuid": "931b23f7-b047-11e8-9af3-005056bb838e",
"name": "0b",
"_links": {
    "self": {
        "href": "/api/network/fc/ports/931b23f7-b047-11e8-9af3-005056bb838e"
    }
}
},
{
"node": {
    "name": "node1",
    "uuid": "3c768e01-1abc-4b3b-b7c0-629ceb62a497",
    "_links": {
        "self": {
            "href": "/api/cluster/nodes/3c768e01-1abc-4b3b-b7c0-629ceb62a497"
        }
    }
}
}

```

```

629ceb62a497"
    }
}
},
"uuid": "931b25ba-b047-11e8-9af3-005056bb838e",
"name": "0c",
"_links": {
    "self": {
        "href": "/api/network/fc/ports/931b25ba-b047-11e8-9af3-005056bb838e"
    }
}
},
{
"node": {
    "name": "node1",
    "uuid": "3c768e01-1abc-4b3b-b7c0-629ceb62a497",
    "_links": {
        "self": {
            "href": "/api/cluster/nodes/3c768e01-1abc-4b3b-b7c0-629ceb62a497"
        }
    }
},
"uuid": "931b2748-b047-11e8-9af3-005056bb838e",
"name": "0d",
"_links": {
    "self": {
        "href": "/api/network/fc/ports/931b2748-b047-11e8-9af3-005056bb838e"
    }
}
},
{
"node": {
    "name": "node1",
    "uuid": "3c768e01-1abc-4b3b-b7c0-629ceb62a497",
    "_links": {
        "self": {
            "href": "/api/cluster/nodes/3c768e01-1abc-4b3b-b7c0-629ceb62a497"
        }
    }
},
"uuid": "931b28c2-b047-11e8-9af3-005056bb838e",
"name": "0e",

```

```

    "_links": {
      "self": {
        "href": "/api/network/fc/ports/931b28c2-b047-11e8-9af3-
005056bb838e"
      }
    }
  },
  {
    "node": {
      "name": "node1",
      "uuid": "3c768e01-1abc-4b3b-b7c0-629ceb62a497",
      "_links": {
        "self": {
          "href": "/api/cluster/nodes/3c768e01-1abc-4b3b-b7c0-
629ceb62a497"
        }
      }
    },
    "uuid": "931b2a7b-b047-11e8-9af3-005056bb838e",
    "name": "0f",
    "_links": {
      "self": {
        "href": "/api/network/fc/ports/931b2a7b-b047-11e8-9af3-
005056bb838e"
      }
    }
  },
  {
    "node": {
      "name": "node1",
      "uuid": "3c768e01-1abc-4b3b-b7c0-629ceb62a497",
      "_links": {
        "self": {
          "href": "/api/cluster/nodes/3c768e01-1abc-4b3b-b7c0-
629ceb62a497"
        }
      }
    },
    "uuid": "931b2e2b-b047-11e8-9af3-005056bb838e",
    "name": "1b",
    "_links": {
      "self": {
        "href": "/api/network/fc/ports/931b2e2b-b047-11e8-9af3-
005056bb838e"
      }
    }
  }
}

```

```

    }
],
"num_records": 8,
"_links": {
  "self": {
    "href": "/api/network/fc/ports"
  }
}
}
}

```

Retrieving all FC ports with state *online*

The `state` query parameter is used to perform the query.

```

# The API:
GET /api/network/fc/ports

# The call:
curl -X GET "https://<mgmt-ip>/api/network/fc/ports?state=online" -H
"accept: application/hal+json"

# The response:
{
"records": [
  {
    "node": {
      "name": "node1",
      "uuid": "3c768e01-1abc-4b3b-b7c0-629ceb62a497",
      "_links": {
        "self": {
          "href": "/api/cluster/nodes/3c768e01-1abc-4b3b-b7c0-
629ceb62a497"
        }
      }
    },
    "uuid": "931b20f8-b047-11e8-9af3-005056bb838e",
    "name": "0a",
    "state": "online",
    "_links": {
      "self": {
        "href": "/api/network/fc/ports/931b20f8-b047-11e8-9af3-
005056bb838e"
      }
    }
  }
]
}

```

```

} ,
{
  "node": {
    "name": "node1",
    "uuid": "3c768e01-1abc-4b3b-b7c0-629ceb62a497",
    "_links": {
      "self": {
        "href": "/api/cluster/nodes/3c768e01-1abc-4b3b-b7c0-629ceb62a497"
      }
    }
  },
  "uuid": "931b23f7-b047-11e8-9af3-005056bb838e",
  "name": "0b",
  "state": "online",
  "_links": {
    "self": {
      "href": "/api/network/fc/ports/931b23f7-b047-11e8-9af3-005056bb838e"
    }
  }
},
{
  "node": {
    "name": "node1",
    "uuid": "3c768e01-1abc-4b3b-b7c0-629ceb62a497",
    "_links": {
      "self": {
        "href": "/api/cluster/nodes/3c768e01-1abc-4b3b-b7c0-629ceb62a497"
      }
    }
  },
  "uuid": "931b25ba-b047-11e8-9af3-005056bb838e",
  "name": "0c",
  "state": "online",
  "_links": {
    "self": {
      "href": "/api/network/fc/ports/931b25ba-b047-11e8-9af3-005056bb838e"
    }
  }
}
],
"num_records": 3,
"_links": {

```

```

  "self": {
    "href": "/api/network/fc/ports?state=online"
  }
}
}

```

Retrieving an FC port

```

# The API:
GET /api/network/fc/ports/{uuid}

# The call:
curl -X GET "https://<mgmt-ip>/api/network/fc/ports/931b20f8-b047-11e8-
9af3-005056bb838e" -H "accept: application/hal+json"

# The response:
{
  "node": {
    "name": "node1",
    "uuid": "5a534a72-b047-11e8-9af3-005056bb838e",
    "_links": {
      "self": {
        "href": "/api/cluster/nodes/5a534a72-b047-11e8-9af3-005056bb838e"
      }
    }
  },
  "uuid": "931b20f8-b047-11e8-9af3-005056bb838e",
  "name": "0a",
  "description": "Fibre Channel Target Adapter 0a (ACME Fibre Channel
Adapter, rev. 1.0.0, 8G)",
  "enabled": true,
  "fabric": {
    "connected": true,
    "connected_speed": 8,
    "name": "55:0e:b1:a0:20:40:80:00",
    "port_address": "52100",
    "switch_port": "ssan-g620-03:1"
  },
  "physical_protocol": "fibre_channel",
  "speed": {
    "maximum": "8",
    "configured": "auto"
  },
  "state": "online",
}

```

```
"supported_protocols": [
    "fcp"
],
"transceiver": {
    "form_factor": "SFP",
    "manufacturer": "ACME",
    "capabilities": [
        4,
        8
    ],
    "part_number": "1000"
},
"wwnn": "50:0a:09:80:bb:83:8e:00",
"wwpn": "50:0a:09:82:bb:83:8e:00",
"metric": {
    "timestamp": "2019-04-09T05:50:15Z",
    "duration": "PT15S",
    "status": "ok",
    "latency": {
        "other": 0,
        "total": 0,
        "read": 0,
        "write": 0
    },
    "iops": {
        "read": 0,
        "write": 0,
        "other": 0,
        "total": 0
    },
    "throughput": {
        "read": 0,
        "write": 0,
        "total": 0
    }
},
"statistics": {
    "timestamp": "2019-04-09T05:50:42Z",
    "status": "ok",
    "latency_raw": {
        "other": 38298,
        "total": 38298,
        "read": 0,
        "write": 0
    },
    "iops_raw": {

```

```

    "read": 0,
    "write": 0,
    "other": 3,
    "total": 3
  },
  "throughput_raw": {
    "read": 0,
    "write": 0,
    "total": 0
  }
},
"_links": {
  "self": {
    "href": "/api/network/fc/ports/931b20f8-b047-11e8-9af3-005056bb838e"
  }
}
}

```

Disabling an FC port

If an active FC interface exists on an FC port, the port cannot be disabled.

```

# The API:
PATCH /api/network/fc/ports/{uuid}

# The call:
curl -X PATCH "http://<mgmt-ip>/api/network/fc/ports/931b20f8-b047-11e8-9af3-005056bb838e" -H "accept: application/hal+json" -d '{ "enabled": false }'

```

Retrieve FC ports

GET /network/fc/ports

Introduced In: 9.6

Retrieves FC ports.

Expensive properties

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

- fabric.name

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

Related ONTAP commands

- `network fcp adapter show`

Learn more

- [DOC /network/fc/ports](#)

Parameters

| Name | Type | In | Required | Description |
|-----------------------------------|---------|-------|----------|--|
| <code>description</code> | string | query | False | Filter by description |
| <code>name</code> | string | query | False | Filter by name |
| <code>metric.latency.total</code> | integer | query | False | Filter by <code>metric.latency.total</code> <ul style="list-style-type: none"> • Introduced in: 9.8 |
| <code>metric.latency.read</code> | integer | query | False | Filter by <code>metric.latency.read</code> <ul style="list-style-type: none"> • Introduced in: 9.8 |
| <code>metric.latency.other</code> | integer | query | False | Filter by <code>metric.latency.other</code> <ul style="list-style-type: none"> • Introduced in: 9.8 |
| <code>metric.latency.write</code> | integer | query | False | Filter by <code>metric.latency.write</code> <ul style="list-style-type: none"> • Introduced in: 9.8 |
| <code>metric.status</code> | string | query | False | Filter by <code>metric.status</code> <ul style="list-style-type: none"> • Introduced in: 9.8 |

| Name | Type | In | Required | Description |
|-------------------------|---------|-------|----------|--|
| metric.iops.total | integer | query | False | Filter by metric.iops.total <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.iops.read | integer | query | False | Filter by metric.iops.read <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.iops.other | integer | query | False | Filter by metric.iops.other <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.iops.write | integer | query | False | Filter by metric.iops.write <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.duration | string | query | False | Filter by metric.duration <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.timestamp | string | query | False | Filter by metric.timestamp <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.throughput.write | integer | query | False | Filter by metric.throughput.write <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.throughput.read | integer | query | False | Filter by metric.throughput.read <ul style="list-style-type: none"> • Introduced in: 9.8 |

| Name | Type | In | Required | Description |
|-------------------------|---------|-------|----------|--|
| metric.throughput.total | integer | query | False | Filter by metric.throughput.total <ul style="list-style-type: none"> • Introduced in: 9.8 |
| uuid | string | query | False | Filter by uuid |
| wwnn | string | query | False | Filter by wwnn |
| wwpn | string | query | False | Filter by wwpn |
| node.name | string | query | False | Filter by node.name |
| node.uuid | string | query | False | Filter by node.uuid |
| physical_protocol | string | query | False | Filter by physical_protocol |
| fabric.name | string | query | False | Filter by fabric.name |
| fabric.port_address | string | query | False | Filter by fabric.port_address |
| fabric.connected | boolean | query | False | Filter by fabric.connected |
| fabric.connected_speed | integer | query | False | Filter by fabric.connected_speed |
| fabric.switch_port | string | query | False | Filter by fabric.switch_port |
| enabled | boolean | query | False | Filter by enabled |
| speed.maximum | string | query | False | Filter by speed.maximum |
| speed.configured | string | query | False | Filter by speed.configured |
| supported_protocols | string | query | False | Filter by supported_protocols |

| Name | Type | In | Required | Description |
|------------------------------|---------|-------|----------|---|
| state | string | query | False | Filter by state |
| statistics.iops_raw.to tal | integer | query | False | Filter by statistics.iops_raw.to tal <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.iops_raw.read | integer | query | False | Filter by statistics.iops_raw.read <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.iops_raw.other | integer | query | False | Filter by statistics.iops_raw.other <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.iops_raw.write | integer | query | False | Filter by statistics.iops_raw.write <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.latency_raw.total | integer | query | False | Filter by statistics.latency_raw.total <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.latency_raw.read | integer | query | False | Filter by statistics.latency_raw.read <ul style="list-style-type: none"> • Introduced in: 9.8 |

| Name | Type | In | Required | Description |
|---------------------------------|---------|-------|----------|--|
| statistics.latency_rw.other | integer | query | False | Filter by statistics.latency_rw.other <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.latency_rw.write | integer | query | False | Filter by statistics.latency_rw.write <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.timestamp | string | query | False | Filter by statistics.timestamp <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.status | string | query | False | Filter by statistics.status <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.throughput_raw.write | integer | query | False | Filter by statistics.throughput_raw.write <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.throughput_raw.read | integer | query | False | Filter by statistics.throughput_raw.read <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.throughput_raw.total | integer | query | False | Filter by statistics.throughput_raw.total <ul style="list-style-type: none"> • Introduced in: 9.8 |

| Name | Type | In | Required | Description |
|--------------------------|---------------|-------|----------|--|
| transceiver.manufacturer | string | query | False | Filter by transceiver.manufacturer |
| transceiver.part_number | string | query | False | Filter by transceiver.part_number |
| transceiver.form-factor | string | query | False | Filter by transceiver.form-factor |
| transceiver.capabilities | integer | query | False | Filter by transceiver.capabilities |
| fields | array[string] | query | False | Specify the fields to return. |
| max_records | integer | query | False | Limit the number of records returned. |
| return_records | boolean | query | False | <p>The default is true for GET calls. When set to false, only the number of records is returned.</p> <ul style="list-style-type: none"> Default value: 1 |
| 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 Max value: 120 Min value: 0 |

| Name | Type | In | Required | Description |
|----------|---------------|-------|----------|--|
| order_by | array[string] | query | False | Order results by specified fields and optional [asc] |

Response

Status: 200, Ok

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

Example response

```
{  
  "_links": {  
    "next": {  
      "href": "/api/resourcelink"  
    },  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "records": [  
    {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "description": "Fibre Channel Target Adapter 0a (ACME Fibre  
Channel Adapter, rev. 1.0.0, 8G)",  
      "fabric": {  
        "connected_speed": "16",  
        "name": "string",  
        "port_address": "52100A",  
        "switch_port": "ssan-g620-03:33"  
      },  
      "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"  
        }  
      }  
    }  
  ]  
}
```

```

        "total": "1000",
        "write": "100"
    },
    "timestamp": "2017-01-25T11:20:13Z"
},
"name": "0a",
"node": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "name": "node1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"physical_protocol": "string",
"speed": {
    "configured": "auto",
    "maximum": "32"
},
"state": "online",
"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": "2017-01-25T11:20:13Z"
},
"supported_protocols": [
    "string"
],
"transceiver": {
    "capabilities": [
        "16"

```

```

        ],
        "form-factor": "string",
        "manufacturer": "Acme, Inc.",
        "part_number": "string"
    },
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",
    "wwnn": "20:00:00:50:56:b4:13:a8",
    "wwpn": "20:00:00:50:56:b4:13:a8"
}
]
}

```

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 | |

fabric

Properties of the fabric to which the FC port is attached.

| Name | Type | Description |
|-----------------|---------|--|
| connected | boolean | Reports if the physical port has established a connection with the FC fabric. |
| connected_speed | integer | The negotiated data rate between the target FC port and the fabric in gigabits per second. |
| name | string | <p>The name of the fabric to which the port is connected. This is only available when the FC port is connected to a fabric.</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 Requesting specific fields to learn more.</p> |

| Name | Type | Description |
|--------------|--------|--|
| port_address | string | <p>The FC port address of the host bus adapter (HBA) physical port.</p> <p>Each FC port in an FC switched fabric has its own unique FC port address for routing purposes. The FC port address is assigned by a switch in the fabric when that port logs in to the fabric. This property refers to the FC port address given to the physical host bus adapter (HBA) port when the port performs a fabric login (FLOGI).</p> <p>This is useful for obtaining statistics and diagnostic information from FC switches.</p> <p>This is a six-digit hexadecimal encoded numeric value.</p> |
| switch_port | string | The switch port to which the FC port is connected. |

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

Performance numbers, such as IOPS latency and throughput

| 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. |

node

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| uuid | string | |

speed

The physical device speed related properties of the FC port.

| Name | Type | Description |
|------------|--------|--|
| configured | string | The configured speed of the FC port in gigabits per second. |
| maximum | string | The maximum speed supported by the FC port in gigabits per second. |

iops_raw

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

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

latency_raw

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

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

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

throughput_raw

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

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

statistics

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

| Name | Type | Description |
|-------------|-------------|--|
| iops_raw | 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. |

| Name | Type | Description |
|----------------|--------------------------------|--|
| 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. |

transceiver

Properties of the transceiver connected to the FC port.

| Name | Type | Description |
|--------------|----------------|--|
| capabilities | array[integer] | The speeds of which the transceiver is capable in gigabits per second. |

| Name | Type | Description |
|--------------|--------|--|
| form-factor | string | <p>The form factor of the transceiver. Possible values are:</p> <ul style="list-style-type: none"> • <i>sfp</i> - Small Form Factor - Pluggable • <i>sff</i> - Small Form Factor • <i>unk</i> - Unknown |
| manufacturer | string | The manufacturer of the transceiver. |
| part_number | string | The part number of the transceiver. |

fc_port

A Fibre Channel (FC) port is the physical port of an FC adapter on an ONTAP cluster node that can be connected to an FC network to provide FC network connectivity. An FC port defines the location of an FC interface within the ONTAP cluster.

| Name | Type | Description |
|-------------------|------------------------|---|
| _links | _links | |
| description | string | A description of the FC port. |
| enabled | boolean | The administrative state of the FC port. If this property is set to <i>false</i> , all FC connectivity to FC interfaces are blocked. Optional in PATCH. |
| fabric | fabric | Properties of the fabric to which the FC port is attached. |
| metric | metric | Performance numbers, such as IOPS latency and throughput |
| name | string | The FC port name. |
| node | node | |
| physical_protocol | string | The physical network protocol of the FC port. |
| speed | speed | The physical device speed related properties of the FC port. |

| Name | Type | Description |
|---------------------|-----------------------------|---|
| state | string | <p>The operational state of the FC port.</p> <ul style="list-style-type: none"> • startup - The port is booting up. • link_not_connected - The port has finished initialization, but a link with the fabric is not established. • online - The port is initialized and a link with the fabric has been established. • link_disconnected - The link was present at one point on this port but is currently not established. • offlined_by_user - The port is administratively disabled. • offlined_by_system - The port is set to offline by the system. This happens when the port encounters too many errors. • node_offline - The state information for the port cannot be retrieved. The node is offline or inaccessible. |
| statistics | statistics | These are raw performance numbers, such as IOPS latency and throughput. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster. |
| supported_protocols | array[string] | The network protocols supported by the FC port. |
| transceiver | transceiver | Properties of the transceiver connected to the FC port. |
| uuid | string | The unique identifier of the FC port. |
| wwnn | string | The base world wide node name (WWNN) for the FC port. |

| Name | Type | Description |
|------|--------|---|
| wwpn | string | The base world wide port name (WWPN) for the FC port. |

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 FC port

GET /network/fc/ports/{uuid}

Introduced In: 9.6

Retrieves an FC port.

Expensive properties

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

- `fabric.name`
- `statistics.*`
- `metric.*`

Related ONTAP commands

- `network fcp adapter show`

Learn more

- [DOC /network/fc/ports](#)

Parameters

| Name | Type | In | Required | Description |
|--------|---------------|-------|----------|--|
| uuid | string | path | True | The unique identifier for the FC port. |
| fields | array[string] | query | False | Specify the fields to return. |

Response

Status: 200, Ok

| Name | Type | Description |
|-------------------|------------------------|---|
| _links | _links | |
| description | string | A description of the FC port. |
| enabled | boolean | The administrative state of the FC port. If this property is set to <code>false</code> , all FC connectivity to FC interfaces are blocked. Optional in PATCH. |
| fabric | fabric | Properties of the fabric to which the FC port is attached. |
| metric | metric | Performance numbers, such as IOPS latency and throughput |
| name | string | The FC port name. |
| node | node | |
| physical_protocol | string | The physical network protocol of the FC port. |
| speed | speed | The physical device speed related properties of the FC port. |

| Name | Type | Description |
|---------------------|-----------------------------|---|
| state | string | <p>The operational state of the FC port.</p> <ul style="list-style-type: none"> • startup - The port is booting up. • link_not_connected - The port has finished initialization, but a link with the fabric is not established. • online - The port is initialized and a link with the fabric has been established. • link_disconnected - The link was present at one point on this port but is currently not established. • offlined_by_user - The port is administratively disabled. • offlined_by_system - The port is set to offline by the system. This happens when the port encounters too many errors. • node_offline - The state information for the port cannot be retrieved. The node is offline or inaccessible. |
| statistics | statistics | These are raw performance numbers, such as IOPS latency and throughput. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster. |
| supported_protocols | array[string] | The network protocols supported by the FC port. |
| transceiver | transceiver | Properties of the transceiver connected to the FC port. |
| uuid | string | The unique identifier of the FC port. |
| wwnn | string | The base world wide node name (WWNN) for the FC port. |
| wwpn | string | The base world wide port name (WWPN) for the FC port. |

Example response

```
{  
  "_links": {  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "description": "Fibre Channel Target Adapter 0a (ACME Fibre Channel  
Adapter, rev. 1.0.0, 8G)",  
  "fabric": {  
    "connected_speed": "16",  
    "name": "string",  
    "port_address": "52100A",  
    "switch_port": "ssan-g620-03:33"  
  },  
  "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": "2017-01-25T11:20:13Z"  
  },  
  "name": "0a",  
  "node": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    }  
  }  
}
```

```
        }
    },
    "name": "node1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"physical_protocol": "string",
"speed": {
    "configured": "auto",
    "maximum": "32"
},
"state": "online",
"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": "2017-01-25T11:20:13Z"
},
"supported_protocols": [
    "string"
],
"transceiver": {
    "capabilities": [
        "16"
    ],
    "form-factor": "string",
    "manufacturer": "Acme, Inc.",
    "part_number": "string"
},
"uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",
"wwnn": "20:00:00:50:56:b4:13:a8",
"wwpn": "20:00:00:50:56:b4:13:a8"
}
```

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 | |

fabric

Properties of the fabric to which the FC port is attached.

| Name | Type | Description |
|-----------------|---------|--|
| connected | boolean | Reports if the physical port has established a connection with the FC fabric. |
| connected_speed | integer | The negotiated data rate between the target FC port and the fabric in gigabits per second. |
| name | string | <p>The name of the fabric to which the port is connected. This is only available when the FC port is connected to a fabric.</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 Requesting specific fields to learn more.</p> |

| Name | Type | Description |
|--------------|--------|--|
| port_address | string | <p>The FC port address of the host bus adapter (HBA) physical port.</p> <p>Each FC port in an FC switched fabric has its own unique FC port address for routing purposes. The FC port address is assigned by a switch in the fabric when that port logs in to the fabric. This property refers to the FC port address given to the physical host bus adapter (HBA) port when the port performs a fabric login (FLOGI).</p> <p>This is useful for obtaining statistics and diagnostic information from FC switches.</p> <p>This is a six-digit hexadecimal encoded numeric value.</p> |
| switch_port | string | The switch port to which the FC port is connected. |

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

Performance numbers, such as IOPS latency and throughput

| 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. |

node

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| uuid | string | |

speed

The physical device speed related properties of the FC port.

| Name | Type | Description |
|------------|--------|--|
| configured | string | The configured speed of the FC port in gigabits per second. |
| maximum | string | The maximum speed supported by the FC port in gigabits per second. |

iops_raw

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

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

latency_raw

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

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

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

throughput_raw

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

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

statistics

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

| Name | Type | Description |
|-------------|-------------|--|
| iops_raw | 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. |

| Name | Type | Description |
|----------------|--------------------------------|--|
| 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. |

transceiver

Properties of the transceiver connected to the FC port.

| Name | Type | Description |
|--------------|----------------|--|
| capabilities | array[integer] | The speeds of which the transceiver is capable in gigabits per second. |

| Name | Type | Description |
|--------------|--------|--|
| form-factor | string | <p>The form factor of the transceiver. Possible values are:</p> <ul style="list-style-type: none"> • <i>sfp</i> - Small Form Factor - Pluggable • <i>sff</i> - Small Form Factor • <i>unk</i> - Unknown |
| manufacturer | string | The manufacturer of the transceiver. |
| part_number | string | The part number of the transceiver. |

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 FC port

PATCH /network/fc/ports/{uuid}

Introduced In: 9.6

Updates an FC port.

Related ONTAP commands

- `network fcp adapter modify`

Learn more

- [DOC /network/fc/ports](#)

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|--|
| uuid | string | path | True | The unique identifier for the FC port. |

Request Body

| Name | Type | Description |
|-------------------|---------|---|
| description | string | A description of the FC port. |
| enabled | boolean | The administrative state of the FC port. If this property is set to <code>false</code> , all FC connectivity to FC interfaces are blocked. Optional in PATCH. |
| fabric | fabric | Properties of the fabric to which the FC port is attached. |
| name | string | The FC port name. |
| node | node | |
| physical_protocol | string | The physical network protocol of the FC port. |
| speed | speed | The physical device speed related properties of the FC port. |

| Name | Type | Description |
|---------------------|-----------------------------|---|
| state | string | <p>The operational state of the FC port.</p> <ul style="list-style-type: none"> • startup - The port is booting up. • link_not_connected - The port has finished initialization, but a link with the fabric is not established. • online - The port is initialized and a link with the fabric has been established. • link_disconnected - The link was present at one point on this port but is currently not established. • offlined_by_user - The port is administratively disabled. • offlined_by_system - The port is set to offline by the system. This happens when the port encounters too many errors. • node_offline - The state information for the port cannot be retrieved. The node is offline or inaccessible. |
| supported_protocols | array[string] | The network protocols supported by the FC port. |
| transceiver | transceiver | Properties of the transceiver connected to the FC port. |
| uuid | string | The unique identifier of the FC port. |
| wwnn | string | The base world wide node name (WWNN) for the FC port. |
| wwpn | string | The base world wide port name (WWPN) for the FC port. |

Example request

```
{  
  "description": "Fibre Channel Target Adapter 0a (ACME Fibre Channel  
Adapter, rev. 1.0.0, 8G)",  
  "fabric": {  
    "connected_speed": "16",  
    "name": "string",  
    "port_address": "52100A",  
    "switch_port": "ssan-g620-03:33"  
  },  
  "name": "0a",  
  "node": {  
    "name": "node1",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "physical_protocol": "string",  
  "speed": {  
    "configured": "auto",  
    "maximum": "32"  
  },  
  "state": "online",  
  "supported_protocols": [  
    "string"  
  ],  
  "transceiver": {  
    "capabilities": [  
      "16"  
    ],  
    "form-factor": "string",  
    "manufacturer": "Acme, Inc.",  
    "part_number": "string"  
  },  
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",  
  "wwnn": "20:00:00:50:56:b4:13:a8",  
  "wwpn": "20:00:00:50:56:b4:13:a8"  
}
```

Response

```
Status: 200, Ok
```

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|--|
| 5374085 | The node where the Fibre Channel port is located is offline. |
| 5374087 | The Fibre Channel port has active Fibre Channel interfaces and cannot be disabled. |

Definitions

See Definitions

href

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

_links

fabric

Properties of the fabric to which the FC port is attached.

| Name | Type | Description |
|-----------------|---------|--|
| connected | boolean | Reports if the physical port has established a connection with the FC fabric. |
| connected_speed | integer | The negotiated data rate between the target FC port and the fabric in gigabits per second. |
| name | string | <p>The name of the fabric to which the port is connected. This is only available when the FC port is connected to a fabric.</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 Requesting specific fields to learn more.</p> |

| Name | Type | Description |
|--------------|--------|--|
| port_address | string | <p>The FC port address of the host bus adapter (HBA) physical port.</p> <p>Each FC port in an FC switched fabric has its own unique FC port address for routing purposes. The FC port address is assigned by a switch in the fabric when that port logs in to the fabric. This property refers to the FC port address given to the physical host bus adapter (HBA) port when the port performs a fabric login (FLOGI).</p> <p>This is useful for obtaining statistics and diagnostic information from FC switches.</p> <p>This is a six-digit hexadecimal encoded numeric value.</p> |
| switch_port | string | The switch port to which the FC port is connected. |

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

Performance numbers, such as IOPS latency and throughput

| 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. |

| 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. |

node

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| uuid | string | |

speed

The physical device speed related properties of the FC port.

| Name | Type | Description |
|------------|--------|--|
| configured | string | The configured speed of the FC port in gigabits per second. |
| maximum | string | The maximum speed supported by the FC port in gigabits per second. |

iops_raw

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

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

latency_raw

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

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

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

throughput_raw

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

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

statistics

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

| Name | Type | Description |
|-------------|-------------|--|
| iops_raw | 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. |

| Name | Type | Description |
|----------------|--------------------------------|--|
| 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. |

transceiver

Properties of the transceiver connected to the FC port.

| Name | Type | Description |
|--------------|----------------|--|
| capabilities | array[integer] | The speeds of which the transceiver is capable in gigabits per second. |

| Name | Type | Description |
|--------------|--------|--|
| form-factor | string | <p>The form factor of the transceiver. Possible values are:</p> <ul style="list-style-type: none"> • <i>sfp</i> - Small Form Factor - Pluggable • <i>sff</i> - Small Form Factor • <i>unk</i> - Unknown |
| manufacturer | string | The manufacturer of the transceiver. |
| part_number | string | The part number of the transceiver. |

fc_port

A Fibre Channel (FC) port is the physical port of an FC adapter on an ONTAP cluster node that can be connected to an FC network to provide FC network connectivity. An FC port defines the location of an FC interface within the ONTAP cluster.

| Name | Type | Description |
|-------------------|---------|---|
| description | string | A description of the FC port. |
| enabled | boolean | The administrative state of the FC port. If this property is set to <i>false</i> , all FC connectivity to FC interfaces are blocked. Optional in PATCH. |
| fabric | fabric | Properties of the fabric to which the FC port is attached. |
| name | string | The FC port name. |
| node | node | |
| physical_protocol | string | The physical network protocol of the FC port. |
| speed | speed | The physical device speed related properties of the FC port. |

| Name | Type | Description |
|---------------------|-----------------------------|---|
| state | string | <p>The operational state of the FC port.</p> <ul style="list-style-type: none"> • startup - The port is booting up. • link_not_connected - The port has finished initialization, but a link with the fabric is not established. • online - The port is initialized and a link with the fabric has been established. • link_disconnected - The link was present at one point on this port but is currently not established. • offlined_by_user - The port is administratively disabled. • offlined_by_system - The port is set to offline by the system. This happens when the port encounters too many errors. • node_offline - The state information for the port cannot be retrieved. The node is offline or inaccessible. |
| supported_protocols | array[string] | The network protocols supported by the FC port. |
| transceiver | transceiver | Properties of the transceiver connected to the FC port. |
| uuid | string | The unique identifier of the FC port. |
| wwnn | string | The base world wide node name (WWNN) for the FC port. |
| wwpn | string | The base world wide port name (WWPN) for the FC port. |

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 FC port historical performance metrics

GET /network/fc/ports/{uuid}/metrics

Introduced In: 9.8

Retrieves historical performance metrics for an FC port

Parameters

| Name | Type | In | Required | Description |
|------------------|---------|-------|----------|----------------------------|
| duration | string | query | False | Filter by duration |
| timestamp | string | query | False | Filter by timestamp |
| 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 |
| latency.total | integer | query | False | Filter by latency.total |

| Name | Type | In | Required | Description |
|---------------|---------|-------|----------|-----------------------------------|
| latency.read | integer | query | False | Filter by latency.read |
| latency.other | integer | query | False | Filter by latency.other |
| latency.write | integer | query | False | Filter by latency.write |
| status | string | query | False | Filter by status |
| iops.total | integer | query | False | Filter by iops.total |
| iops.read | integer | query | False | Filter by iops.read |
| iops.other | integer | query | False | Filter by iops.other |
| iops.write | integer | query | False | Filter by iops.write |
| uuid | string | path | True | Unique identifier of the FC port. |

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

| Name | Type | In | Required | Description |
|--|----------------|---------|----------|--|
| return_timeout | integer | query | False | <p>The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached.</p> <ul style="list-style-type: none"> • Default value: 1 • Max value: 120 • Min value: 0 |
| fields | array[string] | query | False | Specify the fields to return. |
| max_records | integer | query | False | Limit the number of records returned. |
| order_by | array[string] | query | False | Order results by specified fields and optional [asc] |
| desc] direction. Default direction is 'asc' for ascending. | return_records | boolean | query | False |

Response

Status: 200, Ok

| Name | Type | Description |
|-------------|------------------------|-------------------|
| _links | _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": "2017-01-25T11:20:13Z"  
    }  
  ]  
}
```

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. |

records

Performance numbers, such as IOPS latency and throughput

| 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. |

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 HTTP proxy configuration

Network http-proxy endpoint overview

Overview

Configuration of an HTTP proxy for an SVM or a Cluster IPspace.

Retrieve HTTP proxy information

The HTTP proxy GET operation retrieves all configurations for an SVM or a Cluster IPspace via '/api/cluster'.

Examples

Retrieving all fields for all HTTP proxy configurations

```

# The API:
/api/network/http-proxy

# The call:
curl -X GET "https://<mgmt-ip>/api/network/http-
proxy?fields=*&return_records=true&return_timeout=15" -H "accept:
application/json"

# The response:
{
  "records": [
    {
      "uuid": "4133a1fc-7228-11e9-b40c-005056bb4f0c",
      "svm": {
        "name": "vs1",
        "uuid": "4133a1fc-7228-11e9-b40c-005056bb4f0c"
      },
      "server": "server1.example.com",
      "port": 3128
    },
    {
      "uuid": "96219ce3-7214-11e9-828c-005056bb4f0c",
      "svm": {
        "name": "cluster-1",
        "uuid": "96219ce3-7214-11e9-828c-005056bb4f0c"
      },
      "ipspace": {
        "uuid": "7433520f-7214-11e9-828c-005056bb4f0c",
        "name": "Default"
      },
      "server": "1.1.1.",
      "port": 3128
    }
  ],
  "num_records": 2
}

```

Retrieving the HTTP proxy configuration for a specific SVM

```

# The API:
/api/network/http-proxy/{uuid}

# The call:
curl -X GET "https://<mgmt-ip>/api/network/http-proxy/96219ce3-7214-11e9-
828c-005056bb4f0c" -H "accept: application/json"

# The response
{
  "uuid": "96219ce3-7214-11e9-828c-005056bb4f0c",
  "svm": {
    "name": "cluster-1",
    "uuid": "96219ce3-7214-11e9-828c-005056bb4f0c"
  },
  "ipspace": {
    "uuid": "7433520f-7214-11e9-828c-005056bb4f0c",
    "name": "Default"
  },
  "server": "1.1.1.1",
  "port": 3128
}

```

Creating an HTTP proxy configuration

You can use the HTTP proxy POST operation to create an HTTP proxy configuration for the specified SVM.

Examples

Creating an HTTP proxy configuration for a particular SVM

```

# The API:
/api/network/http-proxy

# The call:
curl -X POST "https://<mgmt-ip>/api/network/http-proxy" -H "accept:
application/json" -H "Content-Type: application/json" -d "{      \"port\":
3128,  \"server\": \"1.1.1.1\",  \"svm\": {      \"name\": \"cluster-1\"}
}"

```

Creating an HTTP proxy configuration for a particular IPspace

```
# The API:  
/api/network/http-proxy  
  
# The call:  
curl -X POST "https://<mgmt-ip>/api/network/http-proxy" -H "accept: application/json" -H "Content-Type: application/json" -d "{ \"ipspace\": { \"name\": \"Default\" }, \"port\": 3128, \"server\": \"1.1.1.1\" }"
```

Update an HTTP proxy configuration for a specified SVM

You can use the HTTP proxy PATCH operation to update the HTTP proxy configuration for the specified SVM.

Example

The following example shows how a PATCH operation is used to update an HTTP proxy configuration for a specific SVM:

```
# The API:  
/api/network/http-proxy/{uuid}  
  
# The call:  
curl -X PATCH "https://<mgmt-ip>/api/network/http-proxy/96219ce3-7214-11e9-828c-005056bb4f0c" -H "accept: application/json" -H "Content-Type: application/json" -d "{ \"port\": 3128, \"server\": \"server2.example.com\" }"
```

Delete an HTTP proxy configuration for a specified SVM

You can use the HTTP proxy DELETE operation to delete the HTTP proxy configuration for the specified SVM.

Example

The following example shows how a DELETE operation is used to delete an HTTP proxy configuration for a specific SVM:

```
# The API:  
/api/network/http-proxy/{uuid}  
  
# The call:  
curl -X DELETE "https://<mgmt-ip>/api/network/http-proxy/96219ce3-7214-11e9-828c-005056bb4f0c" -H "accept: application/json"
```

Retrieve HTTP proxy configurations for all SVMs and cluster IPspaces

GET /network/http-proxy

Introduced In: 9.7

Retrieves the HTTP proxy configurations of all the SVMs and Cluster IPspaces.

Related ONTAP commands

- `vserver http-proxy show`

Parameters

| Name | Type | In | Required | Description |
|----------------|---------------|-------|----------|---|
| server | string | query | False | Filter by server |
| port | integer | query | False | Filter by port |
| scope | string | query | False | Filter by scope |
| ipspace.name | string | query | False | Filter by ipspace.name |
| ipspace.uuid | string | query | False | Filter by ipspace.uuid |
| 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 |
| fields | array[string] | query | False | Specify the fields to return. |
| max_records | integer | query | False | Limit the number of records returned. |
| return_records | boolean | query | False | <p>The default is true for GET calls. When set to false, only the number of records is returned.</p> <ul style="list-style-type: none">• Default value: 1 |

| Name | Type | In | Required | Description |
|----------------|---------------|-------|----------|--|
| return_timeout | integer | query | False | <p>The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached.</p> <ul style="list-style-type: none"> • Default value: 1 • Max value: 120 • Min value: 0 |
| order_by | array[string] | query | False | Order results by specified fields and optional [asc] |

Response

Status: 200, Ok

| Name | Type | Description |
|-------------|---|------------------------------|
| _links | _links | |
| num_records | integer | Number of HTTP proxy records |
| records | array[network_http_proxy] | |

Example response

```
{  
  "_links": {  
    "next": {  
      "href": "/api/resourcelink"  
    },  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "records": [  
    {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "ipspace": {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        },  
        "name": "exchange",  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
      },  
      "port": "3128",  
      "scope": "string",  
      "server": "string",  
      "svm": {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        },  
        "name": "svm1",  
        "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
      },  
      "uuid": "string"  
    }  
  ]  
}
```

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 | |

ipspace

Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input.

| Name | Type | Description |
|--------|--------|--------------|
| _links | _links | |
| name | string | IPspace name |
| uuid | string | IPspace UUID |

svm

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

network_http_proxy

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

| Name | Type | Description |
|---------|---------|---|
| ipspace | ipspace | Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input. |
| port | integer | The port number on which the HTTP proxy service is configured on the proxy server. |
| scope | string | Set to "svm" for proxy owned by an SVM. Otherwise, set to "cluster". |
| server | string | The fully qualified domain name (FQDN) or IP address of the proxy server. |
| svm | svm | |
| uuid | string | The UUID that uniquely identifies the HTTP proxy. |

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 HTTP proxy configuration for an SVM or cluster IPspace

POST /network/http-proxy

Introduced In: 9.7

Creates an HTTP proxy configuration for an SVM or a Cluster IPspace. Important notes:

- IPv6 must be enabled if IPv6 family addresses are specified in the "server" field.
- The server and the port combination specified using the "server" and "port" fields is validated during this operation. The validation will fail in the following scenarios:
 - The HTTP proxy service is not configured on the server.
 - The HTTP proxy service is not running on the specified port.
 - The server is unreachable.

Required properties

- SVM-scoped HTTP proxy
 - `svm.uuid` or `svm.name` - Existing SVM in which to create the HTTP proxy.
- Cluster-scoped HTTP proxy
 - `ipspace.uuid` or `ipspace.name` - Existing Cluster IPspace in which to create the HTTP proxy.
- `server` - HTTP proxy server FQDN or IP address.
- `port` - HTTP proxy server port.

Related ONTAP commands

- `vserver http-proxy create`

Parameters

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

Request Body

| Name | Type | Description |
|----------------------|----------------------|---|
| <code>ipspace</code> | <code>ipspace</code> | Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input. |
| <code>port</code> | <code>integer</code> | The port number on which the HTTP proxy service is configured on the proxy server. |

| Name | Type | Description |
|--------|--------|---|
| scope | string | Set to "svm" for proxy owned by an SVM. Otherwise, set to "cluster". |
| server | string | The fully qualified domain name (FQDN) or IP address of the proxy server. |
| svm | svm | |
| uuid | string | The UUID that uniquely identifies the HTTP proxy. |

Example request

```
{
  "ipspace": {
    "name": "exchange",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "port": "3128",
  "scope": "string",
  "server": "string",
  "svm": {
    "name": "svml",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  },
  "uuid": "string"
}
```

Response

```
Status: 201, Created
```

| Name | Type | Description |
|-------------|---------------------------|------------------------------|
| num_records | integer | Number of HTTP proxy records |
| records | array[network_http_proxy] | |

Example response

```
{  
  "records": [  
    {  
      "ipspace": {  
        "name": "exchange",  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
      },  
      "port": "3128",  
      "scope": "string",  
      "server": "string",  
      "svm": {  
        "name": "svm1",  
        "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
      },  
      "uuid": "string"  
    }  
  ]  
}
```

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|--|
| 26214473 | HTTP proxy configuration is not valid. |
| 26214476 | The "IPspace" parameter should not be specified in the SVM context. |
| 26214477 | The specified IPspace does not exist. |
| 23724130 | Cannot use an IPv6 name server address because there are no IPv6 interfaces. |

Definitions

See Definitions

href

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

_links

ipspace

Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input.

| Name | Type | Description |
|------|--------|--------------|
| name | string | IPspace name |
| uuid | string | IPspace UUID |

svm

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

network_http_proxy

| Name | Type | Description |
|---------|---------|---|
| ipspace | ipspace | Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input. |
| port | integer | The port number on which the HTTP proxy service is configured on the proxy server. |
| scope | string | Set to "svm" for proxy owned by an SVM. Otherwise, set to "cluster". |
| server | string | The fully qualified domain name (FQDN) or IP address of the proxy server. |
| svm | svm | |

| Name | Type | Description |
|------|--------|---|
| uuid | string | The UUID that uniquely identifies the HTTP proxy. |

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. |

Deletes the http proxy configuration of the specified svm or cluster ipspace

related ontap commands

- `vserver http-proxy delete`

`DELETE /network/http-proxy/{uuid}`

Introduced In: 9.7

Deletes the HTTP proxy configuration of the specified SVM or Cluster IPspace.

Related ONTAP commands

- `vserver http-proxy delete`

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|-----------------|
| uuid | string | path | True | HTTP proxy UUID |

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

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. |

Displays the http proxy server, port, and ipspace of the specified svm or cluster ipspace

related ontap commands

- vserver http-proxy show

GET /network/http-proxy/{uuid}

Introduced In: 9.7

Displays the HTTP proxy server, port, and IPspace of the specified SVM or Cluster IPspace.

Related ONTAP commands

- vserver http-proxy show

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|-----------------|
| uuid | string | path | True | HTTP proxy UUID |

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

Response

Status: 200, Ok

| Name | Type | Description |
|---------|---------|---|
| _links | _links | |
| ipspace | ipspace | Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input. |
| port | integer | The port number on which the HTTP proxy service is configured on the proxy server. |
| scope | string | Set to "svm" for proxy owned by an SVM. Otherwise, set to "cluster". |
| server | string | The fully qualified domain name (FQDN) or IP address of the proxy server. |
| svm | svm | |
| uuid | string | The UUID that uniquely identifies the HTTP proxy. |

Example response

```
{  
  "_links": {  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "ipspace": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    },  
    "name": "exchange",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "port": "3128",  
  "scope": "string",  
  "server": "string",  
  "svm": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    },  
    "name": "svml1",  
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
  },  
  "uuid": "string"  
}
```

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 | |

ipspace

Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input.

| Name | Type | Description |
|--------|--------|--------------|
| _links | _links | |
| name | string | IPspace name |
| uuid | string | IPspace UUID |

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 |

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

Updates the proxy server, port, username, and password parameters

important notes: * ipv6 must be enabled if ipv6 family addresses are specified in the "server" field * the server and the port combination specified using the "server" and "port" fields is validated during this operation the validation will fail in the following scenarios: * the http proxy service is not configured on the server * the http proxy service is not running on the specified port * the server is unreachable # related ontap commands * vserver http-proxy modify

PATCH /network/http-proxy/{uuid}

Introduced In: 9.7

Updates the proxy server, port, username, and password parameters. Important notes:

- IPv6 must be enabled if IPv6 family addresses are specified in the "server" field.
- The server and the port combination specified using the "server" and "port" fields is validated during this operation. The validation will fail in the following scenarios:
 - The HTTP proxy service is not configured on the server.
 - The HTTP proxy service is not running on the specified port.
 - The server is unreachable.

Related ONTAP commands

- vserver http-proxy modify

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|-----------------|
| uuid | string | path | True | HTTP proxy UUID |

Request Body

| Name | Type | Description |
|--------|---------|--|
| port | integer | The port number on which the HTTP proxy service is configured on the proxy server. |
| scope | string | Set to "svm" for proxy owned by an SVM. Otherwise, set to "cluster". |
| server | string | The fully qualified domain name (FQDN) or IP address of the proxy server. |
| uuid | string | The UUID that uniquely identifies the HTTP proxy. |

Example request

```
{
  "port": "3128",
  "scope": "string",
  "server": "string",
  "uuid": "string"
}
```

Response

```
Status: 200, Ok
```

Error

```
Status: Default
```

ONTAP Error Response Codes

| Error Code | Description |
|------------|--|
| 26214473 | The HTTP proxy configuration is not valid. |
| 23724130 | Cannot use an IPv6 name server address because there are no IPv6 interfaces. |

Definitions

See Definitions

href

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

_links

ipspace

Applies to both SVM and cluster-scoped objects. Either the UUID or name is supplied on input.

| Name | Type | Description |
|------|--------|--------------|
| name | string | IPspace name |
| uuid | string | IPspace UUID |

svm

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

network_http_proxy

| Name | Type | Description |
|--------|---------|--|
| port | integer | The port number on which the HTTP proxy service is configured on the proxy server. |
| scope | string | Set to "svm" for proxy owned by an SVM. Otherwise, set to "cluster". |
| server | string | The fully qualified domain name (FQDN) or IP address of the proxy server. |
| uuid | string | The UUID that uniquely identifies the HTTP proxy. |

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 BGP peer groups

Network IP BGP peer-groups endpoint overview

Overview

The following operations are supported:

- Creation: POST network/ip/bgp/peer-groups
- Collection Get: GET network/ip/bgp/peer-groups
- Instance Get: GET network/ip/bgp/peer-groups/{uuid}
- Instance Patch: PATCH network/ip/bgp/peer-groups/{uuid}
- Instance Delete: DELETE network/ip/bgp/peer-groups/{uuid}

Retrieving network BGP sessions information

The IP BGP peer-groups GET API retrieves and displays relevant information pertaining to the BGP peer-groups configured in the cluster. The response can contain a list of multiple BGP peer-groups or a specific peer-group. Each BGP peer-group represents a BGP session configured between a local interface and a peer router.

Examples

Retrieving all BGP peer-groups in the cluster

The following example shows the list of all BGP peer-groups configured in a cluster.

```

# The API:
/api/network/ip/bgp/peer-groups

# The call:
curl -X GET "https://<mgmt-ip>/api/network/ip/bgp/peer-groups" -H "accept: application/hal+json"

# The response:
{
  "records": [
    {
      "uuid": "5f22ae9d-87b2-11e9-a3a6-005056bb81a4",
      "name": "pg1",
      "_links": {
        "self": {
          "href": "/api/network/ip/bgp/peer-groups/5f22ae9d-87b2-11e9-a3a6-005056bb81a4"
        }
      }
    },
    {
      "uuid": "5fd08be3-87b2-11e9-952f-005056bb2170",
      "name": "pg2",
      "_links": {
        "self": {
          "href": "/api/network/ip/bgp/peer-groups/5fd08be3-87b2-11e9-952f-005056bb2170"
        }
      }
    }
  ],
  "num_records": 2,
  "_links": {
    "self": {
      "href": "/api/network/ip/bgp/peer-groups"
    }
  }
}

```

Retrieving a specific BGP peer-group

The following example shows the response when a specific BGP peer-group is requested. The system returns an error when there is no peer-group with the requested UUID.

```

# The API:
/api/network/ip/bgp/peer-groups/{uuid}

# The call:
curl -X GET "https://<mgmt-ip>/api/network/ip/bgp/peer-groups/5fd08be3-87b2-11e9-952f-005056bb2170" -H "accept: application/hal+json"

# The response:
{
  "uuid": "5fd08be3-87b2-11e9-952f-005056bb2170",
  "name": "pg2",
  "ipspace": {
    "uuid": "84fd3375-879a-11e9-a3a6-005056bb81a4",
    "name": "Default",
    "_links": {
      "self": {
        "href": "/api/network/ipspaces/84fd3375-879a-11e9-a3a6-005056bb81a4"
      }
    }
  },
  "local": {
    "interface": {
      "uuid": "5e76a305-87b2-11e9-952f-005056bb2170",
      "name": "bgp2",
      "ip": {
        "address": "10.10.10.2"
      }
    },
    "port": {
      "uuid": "f8ff73de-879a-11e9-952f-005056bb2170",
      "name": "e0h",
      "node": {
        "name": "node1"
      }
    }
  },
  "peer": {
    "address": "10.10.10.1",
    "asn": 65501
  },
  "state": "up",
  "_links": {
    "self": {
      "href": "/api/network/ip/bgp/peer-groups/5fd08be3-87b2-11e9-952f-005056bb2170"
    }
  }
}

```

```
    }
}
}
```

Retrieving specific fields and limiting the output using filters

The following example shows the response when a filter is applied (location.port.node.name=node1) and only certain fields are requested. Filtered fields are in the output in addition to the default fields and requested fields.

```
# The API:
/api/network/ip/bgp/peer-groups

# The call:
curl -X GET "https://<mgmt-ip>/api/network/ip/bgp/peer-
groups?local.port.node.name=node1&fields=local.interface.ip,peer" -H
"accept: application/hal+json"

# The response:
{
  "records": [
    {
      "uuid": "5f22ae9d-87b2-11e9-a3a6-005056bb81a4",
      "name": "pg1",
      "local": {
        "interface": {
          "ip": {
            "address": "10.10.10.1"
          }
        },
        "port": {
          "node": {
            "name": "node1"
          }
        }
      },
      "peer": {
        "address": "10.10.10.2",
        "asn": 65501
      },
      "_links": {
        "self": {
          "href": "/api/network/ip/bgp/peer-groups/5f22ae9d-87b2-11e9-a3a6-
```

```
005056bb81a4"
    }
}
}
],
"num_records": 1,
"_links": {
  "self": {
    "href": "/api/network/ip/bgp/peer-
groups?local.port.node.name=node1&fields=local.interface.ip,peer"
  }
}
}
```

Creating a BGP peer-group

The BGP peer-group POST API is used to create a peer-group as shown in the following examples.

Examples

Creating a BGP peer-group with an existing interface

The following example shows how to create a BGP peer-group between an existing interface "bgp1" and peer router with the address "10.10.10.10". The local interface "bgp1" needs to support the management-bgp service, otherwise the system returns an error.

```

# The API:
/api/network/ip/bgp/peer-groups

# The call:
curl -X POST "https://<mgmt-ip>/api/network/ip/bgp/peer-
groups?return_records=true" -d'{"name": "newPg", "ipspace.name": "Default",
"local.interface.name": "bgp1", "peer.address": "10.10.10.10"}'

# The response:
{
  "num_records": 1,
  "records": [
    {
      "uuid": "e3faacc6-87cb-11e9-a3a6-005056bb81a4",
      "name": "newPg",
      "ipspace": {
        "name": "Default"
      },
      "local": {
        "interface": {
          "name": "bgp1"
        }
      },
      "peer": {
        "address": "10.10.10.10"
      },
      "_links": {
        "self": {
          "href": "/api/network/ip/bgp/peer-groups/e3faacc6-87cb-11e9-a3a6-
005056bb81a4"
        }
      }
    }
  ]
}

```

Creating a BGP peer-group and provisioning a new local interface

The following example shows how to create a BGP peer-group with any local interface. If the local interface doesn't exist, the system will create it first before creating the peer-group.

```

# The API:
/api/network/ip/bgp/peer-groups

# The call:
curl -X POST "https://<mgmt-ip>/api/network/ip/bgp/peer-
groups?return_records=true" -d'{"name": "newPg1",
"ipspace.name": "Default", "local": {"interface": {"name": "newlif"}, "ip": {"address": "9.9.9.9", "netmask": "24"}, "port": {"name": "e0f", "node": {"name": "node1"}}, "peer.address": "10.10.10.10"}'

# The response:
{
  "num_records": 1,
  "records": [
    {
      "uuid": "c292f069-8872-11e9-a3a6-005056bb81a4",
      "name": "newPg1",
      "ipspace": {
        "name": "Default"
      },
      "local": {
        "interface": {
          "name": "newlif"
        },
        "port": {
          "name": "e0f",
          "node": {
            "name": "node1"
          }
        }
      },
      "peer": {
        "address": "10.10.10.10"
      },
      "_links": {
        "self": {
          "href": "/api/network/ip/bgp/peer-groups/c292f069-8872-11e9-a3a6-
005056bb81a4"
        }
      }
    }
  ]
}

```

Updating BGP peer-groups

The BGP peer-groups PATCH API is used to update attributes of a peer-group.

Examples

Updating the peer router address

The following example shows how the PATCH request changes the peer router IP address.

```
# The API:  
/api/network/ip/bgp/peer-groups/{uuid}  
  
# The call:  
curl -X PATCH "https://<mgmt-ip>/api/network/ip/bgp/peer-groups/80d271c9-  
1f43-11e9-803e-005056a7646a" -H "accept: application/hal+json" -d  
'{"peer.address": "10.10.10.20" }'  
{  
}
```

Updating the peer-group to a new name

The following example shows how the PATCH request renames the peer-group.

```
# The API:  
/api/network/ip/bgp/peer-groups/{uuid}  
  
# The call:  
curl -X PATCH "https://<mgmt-ip>/api/network/ip/bgp/peer-groups/80d271c9-  
1f43-11e9-803e-005056a7646a" -H "accept: application/hal+json" -d  
'{"name": "NewName" }'  
{  
}
```

Deleting BGP peer-groups

The BGP peer-groups DELETE API is used to delete an BGP peer-group.

Example

Deleting a BGP peer-group

The following DELETE request deletes a BGP peer-group.

```
# The API:  
/api/network/ip/bgp/peer-group/{uuid}  
  
# The call:  
curl -X DELETE "https://<mgmt-ip>/api/network/ip/bgp/peer-groups/80d271c9-  
1f43-11e9-803e-005056a7646a"  
{  
}
```

Retrieve all BGP peer group details for VIP

GET /network/ip/bgp/peer-groups

Introduced In: 9.7

Retrieves the details of all BGP peer groups for VIP.

Related ONTAP Commands

- `network bgp peer-group show`

Parameters

| Name | Type | In | Required | Description |
|----------------------|--------|-------|----------|--------------------------------|
| state | string | query | False | Filter by state |
| local.port.name | string | query | False | Filter by local.port.name |
| local.port.node.name | string | query | False | Filter by local.port.node.name |
| local.port.uuid | string | query | False | Filter by local.port.uuid |
| local.interface.name | string | query | False | Filter by local.interface.name |

| Name | Type | In | Required | Description |
|----------------------------|---------------|-------|----------|---|
| local.interface.uuid | string | query | False | Filter by local.interface.uuid |
| local.interface.ip.address | string | query | False | Filter by local.interface.ip.address |
| uuid | string | query | False | Filter by uuid |
| peer.asn | integer | query | False | Filter by peer.asn |
| peer.address | string | query | False | Filter by peer.address |
| name | string | query | False | Filter by name |
| ipspace.name | string | query | False | Filter by ipspace.name |
| ipspace.uuid | string | query | False | Filter by ipspace.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 | <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 |

| Name | Type | In | Required | Description |
|----------------|---------------|-------|----------|--|
| return_timeout | integer | query | False | <p>The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached.</p> <ul style="list-style-type: none"> • Default value: 1 • Max value: 120 • Min value: 0 |
| order_by | array[string] | query | False | Order results by specified fields and optional [asc] |

Response

Status: 200, Ok

| Name | Type | Description |
|-------------|---|-------------|
| _links | _links | |
| num_records | integer | |
| records | array[bgp_peer_group] | |

Example response

```
{  
  "_links": {  
    "next": {  
      "href": "/api/resourcelink"  
    },  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "num_records": "1",  
  "records": [  
    {  
      "ipspace": {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        },  
        "name": "exchange",  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
      },  
      "local": {  
        "interface": {  
          "_links": {  
            "self": {  
              "href": "/api/resourcelink"  
            }  
          },  
          "ip": {  
            "address": "10.10.10.7"  
          },  
          "name": "lif1",  
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        },  
        "port": {  
          "_links": {  
            "self": {  
              "href": "/api/resourcelink"  
            }  
          },  
          "name": "elb",  
          "node": {  
            "name": "node1"  
          },  
          "port": "12345",  
          "type": "elb"  
        }  
      }  
    }  
  ]  
}
```

```

        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
},
"name": "bgpv4peer",
"peer": {
    "address": "10.10.10.7"
},
"state": "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 | |

ipspace

Either the UUID or name is supplied on input.

| Name | Type | Description |
|--------|--------|--------------|
| _links | _links | |
| name | string | IPspace name |
| uuid | string | IPspace UUID |

ip

IP information

| Name | Type | Description |
|---------|--------|----------------------|
| address | string | IPv4 or IPv6 address |

interface

| Name | Type | Description |
|--------|--------|----------------------------|
| _links | _links | |
| ip | ip | IP information |
| name | string | The name of the interface. |

| Name | Type | Description |
|------|--------|--|
| uuid | string | The UUID that uniquely identifies the interface. |

ip

IP information to create a new interface.

| Name | Type | Description |
|---------|--------|---|
| address | string | IPv4 or IPv6 address |
| netmask | string | Input as netmask length (16) or IPv4 mask (255.255.0.0). For IPv6, you must set the netmask length. The default value is 64. Output is always netmask length. |

node

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

port

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| node | node | |
| uuid | string | |

local

Information describing the local interface that is being used to peer with a router using BGP. On a POST operation, an existing BGP interface is used by specifying the interface, or create a new one by specifying the port and IP address.

| Name | Type | Description |
|-----------|---------------------------|-------------|
| interface | interface | |
| port | port | |

peer

Information describing the router to peer with

| Name | Type | Description |
|---------|---------|----------------------------------|
| address | string | Peer router address |
| asn | integer | Autonomous system number of peer |

bgp_peer_group

A BGP peer group between a local network interface and a router, for the purpose of announcing VIP interface locations for SVMs in this IPspace.

| Name | Type | Description |
|---------|---------|---|
| ipspace | ipspace | Either the UUID or name is supplied on input. |
| local | local | Information describing the local interface that is being used to peer with a router using BGP. On a POST operation, an existing BGP interface is used by specifying the interface, or create a new one by specifying the port and IP address. |
| name | string | Name of the peer group |
| peer | peer | Information describing the router to peer with |
| state | string | State of the peer group |
| uuid | string | UUID of the peer group |

error_arguments

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

error

| Name | Type | Description |
|-----------|------------------------|-------------------|
| arguments | array[error_arguments] | Message arguments |

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

Create a new BGP peer group for VIP

POST /network/ip/bgp/peer-groups

Introduced In: 9.7

Creates a new BGP peer group for VIP. Multipath-routing is turned on cluster-wide automatically if the peer group being created results in multiple paths being available for an existing or future VIP interface.

Required properties

- `name` - Name of the peer-group to create.
- `ipspace.name` or `ipspace.uuid`
 - Required with `local.interface.name` to identify a local interface
 - Optional when `local.interface.uuid` is specified
- `local.interface.uuid` or `local.interface.name`
 - Required when specifying an existing local interface.
- `local.interface.name`, `local.ip` and `local.port`
 - Required to create a new local interface.
- `peer.address` - IP address of the peer router

Related ONTAP commands

- `network bgp peer-group create`

Parameters

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

Request Body

| Name | Type | Description |
|---------|---------|---|
| ipspace | ipspace | Either the UUID or name is supplied on input. |
| local | local | Information describing the local interface that is being used to peer with a router using BGP. On a POST operation, an existing BGP interface is used by specifying the interface, or create a new one by specifying the port and IP address. |
| name | string | Name of the peer group |
| peer | peer | Information describing the router to peer with |
| state | string | State of the peer group |
| uuid | string | UUID of the peer group |

Example request

```
{  
  "ipspace": {  
    "name": "exchange",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "local": {  
    "interface": {  
      "name": "lif1",  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    },  
    "ip": {  
      "address": "10.10.10.7",  
      "netmask": "24"  
    },  
    "port": {  
      "name": "elb",  
      "node": {  
        "name": "node1"  
      },  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    }  
  },  
  "name": "bgpv4peer",  
  "peer": {  
    "address": "10.10.10.7"  
  },  
  "state": "string",  
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
}
```

Response

Status: 201, Created

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|--|
| 1376963 | Duplicate IP address is specified. |
| 1966133 | Since masking an address with a netmask represents an entire IP subnet, the masked and unmasked IP addresses cannot be the same. |
| 1966267 | IPv6 addresses must have a prefix length of 64. |
| 1966269 | IPv4 addresses must have a netmask length between 1 and 32. |
| 1967082 | IPspace name and UUID must match if both are given. |
| 1967155 | The specified local.port.name does not match the location.port.name for the specified local.interface. |
| 1967156 | The specified local.port.node.name does not match the location.port.node.name for the specified local.interface. |
| 1967157 | The specified local.port.uuid does not match the location.port.uuid for the specified local.interface. |
| 1967158 | The specified local.interface.name does not exist in the associated IPspace. local.ip.address and local.ip.netmask are required to create a new LIF. |
| 1967159 | local.interface does not support management-bgp service. |
| 1967160 | The specified local.interface.name does not match the specified interface name of local.interface.uuid. |
| 1967161 | The specified local.interface.uuid does not exist in the specified IPspace. |
| 1967162 | Either local.interface or local.ip and local.port are required to specify a local LIF. |
| 1967163 | The specified local.port.name does not match the specified port name of local.port.uuid. |
| 1967164 | The specified local.port.node.name does not match the specified node name of local.port.uuid. |
| 1967165 | The specified local.port does not exist. |
| 1967166 | ipspace.uuid or ipspace.name must be provided with local.interface.name together to identify a LIF. |
| 1967167 | Internal error. Failed to update BGP configuration for node. Retry the command, if necessary. |
| 1967168 | Internal error. Failed to create a VIP port for IPspace on node. Retry the command, if necessary. |
| 1967169 | Internal error. BGP configuration changed during the operation. Retry the command, if necessary. |

| Error Code | Description |
|------------|---|
| 1967170 | Internal error. VIP port configuration changed during the operation. Retry the command, if necessary. |
| 1967171 | Internal error. Fail to access or update BGP peer group. Retry the command, if necessary. |
| 1967172 | Peer group could not be updated because IPspace does not exist. Retry the command, if necessary. |
| 1967173 | The specified local.ip.address does not match the address for the specified local.interface. |
| 1967174 | The specified local.ip.netmask does not match the netmask for the specified local.interface. |
| 1967176 | The specified local.interface.name does not exist in the associated IPspace. local.port.name, local.port.node.name, or local.port.uuid is required to create a new LIF. |
| 1967177 | Internal error. Failed to access the local interface. Retry the command, if necessary. |
| 1967178 | The IPv6 address specified with local.ip.address is not supported because it is link-local, multicast, v4-compatible, v4-mapped, loopback or "::". |
| 1967179 | The IPv4 address specified with local.ip.address is not supported because it is multicast, loopback or 0.0.0.0. |
| 53281985 | Internal error. Failed to update BGP peer group because BGP LIF moved during the operation. Wait a few minutes and try the command again. |
| 53282006 | BGP peer group could not be updated to use a peer address because the value provided is not a valid peer address. If necessary, try the command again with a routable host address. |
| 53282007 | BGP peer group could not be updated to use a peer address because the address represents a different address family to the address of the associated BGP LIF. If necessary, try the command again with a matching address family. |
| 53282018 | Failed to create BGP peer group because an existing peer group has already established a BGP session between LIF and peer address. If necessary, try the command again with a different BGP LIF or a different peer address. |

Definitions

See Definitions

href

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

_links

ipspace

Either the UUID or name is supplied on input.

| Name | Type | Description |
|------|--------|--------------|
| name | string | IPspace name |
| uuid | string | IPspace UUID |

ip

IP information

interface

| Name | Type | Description |
|------|--------|--|
| ip | ip | IP information |
| name | string | The name of the interface. |
| uuid | string | The UUID that uniquely identifies the interface. |

ip

IP information to create a new interface.

| Name | Type | Description |
|---------|--------|---|
| address | string | IPv4 or IPv6 address |
| netmask | string | Input as netmask length (16) or IPv4 mask (255.255.0.0). For IPv6, you must set the netmask length. The default value is 64. Output is always netmask length. |

node

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

port

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| node | node | |
| uuid | string | |

local

Information describing the local interface that is being used to peer with a router using BGP. On a POST operation, an existing BGP interface is used by specifying the interface, or create a new one by specifying the port and IP address.

| Name | Type | Description |
|-----------|-----------|---|
| interface | interface | |
| ip | ip | IP information to create a new interface. |
| port | port | |

peer

Information describing the router to peer with

| Name | Type | Description |
|---------|---------|----------------------------------|
| address | string | Peer router address |
| asn | integer | Autonomous system number of peer |

bgp_peer_group

A BGP peer group between a local network interface and a router, for the purpose of announcing VIP interface locations for SVMs in this IPspace.

| Name | Type | Description |
|---------|---------|---|
| ipspace | ipspace | Either the UUID or name is supplied on input. |

| Name | Type | Description |
|-------|--------|---|
| local | local | Information describing the local interface that is being used to peer with a router using BGP. On a POST operation, an existing BGP interface is used by specifying the interface, or create a new one by specifying the port and IP address. |
| name | string | Name of the peer group |
| peer | peer | Information describing the router to peer with |
| state | string | State of the peer group |
| uuid | string | UUID of the peer group |

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 a BGP peer group for VIP

DELETE /network/ip/bgp/peer-groups/{uuid}

Introduced In: 9.7

Deletes a BGP peer group for VIP.

Related ONTAP commands

- `network bgp peer-group delete`

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|------------------------|
| uuid | string | path | True | UUID of the peer group |

Response

```
Status: 200, Ok
```

Error

```
Status: Default
```

ONTAP Error Response Codes

| Error Code | Description |
|------------|--|
| 53282019 | Internal error. Failed to remove BGP peer group on node. Wait a few minutes and try the command again. |

| 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 details of a BGP peer group for VIP

GET /network/ip/bgp/peer-groups/{uuid}

Introduced In: 9.7

Retrieves details of a BGP peer group for VIP.

Related ONTAP commands

- `network bgp peer-group show`

Parameters

| Name | Type | In | Required | Description |
|--------|---------------|-------|----------|-------------------------------|
| uuid | string | path | True | UUID of the peer group |
| fields | array[string] | query | False | Specify the fields to return. |

Response

Status: 200, Ok

| Name | Type | Description |
|---------|-------------------------|---|
| ipspace | ipspace | Either the UUID or name is supplied on input. |
| local | local | Information describing the local interface that is being used to peer with a router using BGP. On a POST operation, an existing BGP interface is used by specifying the interface, or create a new one by specifying the port and IP address. |
| name | string | Name of the peer group |
| peer | peer | Information describing the router to peer with |
| state | string | State of the peer group |
| uuid | string | UUID of the peer group |

Example response

```
{  
  "ipspace": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    },  
    "name": "exchange",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "local": {  
    "interface": {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "ip": {  
        "address": "10.10.10.7"  
      },  
      "name": "lif1",  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    },  
    "port": {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "name": "elb",  
      "node": {  
        "name": "node1"  
      },  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    }  
  "name": "bgpv4peer",  
  "peer": {  
    "address": "10.10.10.7"  
  },  
  "state": "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 | |

ipspace

Either the UUID or name is supplied on input.

| Name | Type | Description |
|--------|--------|--------------|
| _links | _links | |
| name | string | IPspace name |
| uuid | string | IPspace UUID |

ip

IP information

| Name | Type | Description |
|---------|--------|----------------------|
| address | string | IPv4 or IPv6 address |

interface

| Name | Type | Description |
|--------|--------|--|
| _links | _links | |
| ip | ip | IP information |
| name | string | The name of the interface. |
| uuid | string | The UUID that uniquely identifies the interface. |

ip

IP information to create a new interface.

| Name | Type | Description |
|---------|--------|---|
| address | string | IPv4 or IPv6 address |
| netmask | string | Input as netmask length (16) or IPv4 mask (255.255.0.0). For IPv6, you must set the netmask length. The default value is 64. Output is always netmask length. |

node

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

port

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| node | node | |
| uuid | string | |

local

Information describing the local interface that is being used to peer with a router using BGP. On a POST operation, an existing BGP interface is used by specifying the interface, or create a new one by specifying the port and IP address.

| Name | Type | Description |
|-----------|---------------------------|-------------|
| interface | interface | |
| port | port | |

peer

Information describing the router to peer with

| Name | Type | Description |
|---------|---------|----------------------------------|
| address | string | Peer router address |
| asn | integer | Autonomous system number of peer |

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 a BGP peer group for VIP

PATCH /network/ip/bgp/peer-groups/{uuid}

Introduced In: 9.7

Updates a BGP peer group for VIP.

Related ONTAP commands

- `network bgp peer-group modify`
- `network bgp peer-group rename`

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|------------------------|
| uuid | string | path | True | UUID of the peer group |

Request Body

| Name | Type | Description |
|-------|--------|---|
| local | local | Information describing the local interface that is being used to peer with a router using BGP. On a POST operation, an existing BGP interface is used by specifying the interface, or create a new one by specifying the port and IP address. |
| name | string | Name of the peer group |
| peer | peer | Information describing the router to peer with |
| state | string | State of the peer group |
| uuid | string | UUID of the peer group |

Example request

```
{
  "name": "bgpv4peer",
  "peer": {
    "address": "10.10.10.7"
  },
  "state": "string",
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
```

Response

```
Status: 200, Ok
```

Error

```
Status: Default
```

ONTAP Error Response Codes

| Error Code | Description |
|------------|---|
| 1967171 | Internal error. Fail to access or update BGP peer group. Retry the command, if necessary. |

| Error Code | Description |
|------------|---|
| 53281998 | Failed to rename the BGP peer group because that name is already assigned to a different peer group in the IPspace. |
| 53282006 | BGP peer group could not be updated to use a peer address because the value provided is not a valid peer address. If necessary, try the command again with a routable host address. |
| 53282007 | BGP peer group could not be updated to use a peer address because the address represents a different address family to the address of the associated BGP LIF. If necessary, try the command again with a matching address family. |
| 53282018 | Failed to create BGP peer group because an existing peer group has already established a BGP session between LIF and peer address. If necessary, try the command again with a different BGP LIF or a different peer address. |

Definitions

See Definitions

href

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

_links

ipspace

Either the UUID or name is supplied on input.

| Name | Type | Description |
|------|--------|--------------|
| name | string | IPspace name |
| uuid | string | IPspace UUID |

ip

IP information

interface

| Name | Type | Description |
|------|--------|--|
| ip | ip | IP information |
| name | string | The name of the interface. |
| uuid | string | The UUID that uniquely identifies the interface. |

ip

IP information to create a new interface.

| Name | Type | Description |
|---------|--------|---|
| address | string | IPv4 or IPv6 address |
| netmask | string | Input as netmask length (16) or IPv4 mask (255.255.0.0). For IPv6, you must set the netmask length. The default value is 64. Output is always netmask length. |

node

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

port

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| node | node | |
| uuid | string | |

local

Information describing the local interface that is being used to peer with a router using BGP. On a POST operation, an existing BGP interface is used by specifying the interface, or create a new one by specifying the port and IP address.

peer

Information describing the router to peer with

| Name | Type | Description |
|---------|--------|---------------------|
| address | string | Peer router address |

bgp_peer_group

A BGP peer group between a local network interface and a router, for the purpose of announcing VIP interface locations for SVMs in this IPspace.

| Name | Type | Description |
|-------|--------|---|
| local | local | Information describing the local interface that is being used to peer with a router using BGP. On a POST operation, an existing BGP interface is used by specifying the interface, or create a new one by specifying the port and IP address. |
| name | string | Name of the peer group |
| peer | peer | Information describing the router to peer with |
| state | string | State of the peer group |

| Name | Type | Description |
|------|--------|------------------------|
| uuid | string | UUID of the peer group |

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 network IP interfaces

Network IP interfaces endpoint overview

Overview

The following operations are supported:

- Creation: POST network/ip/interfaces
- Collection Get: GET network/ip/interfaces
- Instance Get: GET network/ip/interfaces/{uuid}
- Instance Patch: PATCH network/ip/interfaces/{uuid}
- Instance Delete: DELETE network/ip/interfaces/{uuid}

Retrieving network interface information

The IP interfaces GET API retrieves and displays relevant information pertaining to the interfaces configured in the cluster. The response can contain a list of multiple interfaces or a specific interface. The fields returned in the response vary for different interfaces and configurations.

Examples

Retrieving all interfaces in the cluster

The following example shows the list of all interfaces configured in a cluster.

```
# The API:  
/api/network/ip/interfaces  
  
# The call:  
curl -X GET "https://<mgmt-ip>/api/network/ip/interfaces" -H "accept:  
application/hal+json"  
  
# The response:  
{  
  "records": [  
    {  
      "uuid": "14531286-59fc-11e8-ba55-005056b4340f",  
      "name": "user-cluster-01_mgmt1",  
      "_links": {  
        "self": {  
          "href": "/api/network/ip/interfaces/14531286-59fc-11e8-ba55-  
005056b4340f"  
        }  
      }  
    },  
    {  
      "uuid": "145318ba-59fc-11e8-ba55-005056b4340f",  
      "name": "user-cluster-01_clus2",  
      "_links": {  
        "self": {  
          "href": "/api/network/ip/interfaces/145318ba-59fc-11e8-ba55-  
005056b4340f"  
        }  
      }  
    },  
    {  
      "uuid": "14531e45-59fc-11e8-ba55-005056b4340f",  
      "name": "user-cluster-01_clus1",  
      "_links": {  
        "self": {  
          "href": "/api/network/ip/interfaces/14531e45-59fc-11e8-ba55-  
005056b4340f"  
        }  
      }  
    }  
  ]  
}
```

```

} ,
{
  "uuid": "245979de-59fc-11e8-ba55-005056b4340f",
  "name": "cluster_mgmt",
  "_links": {
    "self": {
      "href": "/api/network/ip/interfaces/245979de-59fc-11e8-ba55-005056b4340f"
    }
  }
} ,
{
  "uuid": "c670707c-5a11-11e8-8fcb-005056b4340f",
  "name": "lif1",
  "_links": {
    "self": {
      "href": "/api/network/ip/interfaces/c670707c-5a11-11e8-8fcb-005056b4340f"
    }
  }
}
],
"num_records": 5,
"_links": {
  "self": {
    "href": "/api/network/ip/interfaces"
  }
}
}

```

Retrieving a specific Cluster-scoped interface

The following example shows the response when a specific Cluster-scoped interface is requested. The system returns an error when there is no interface with the requested UUID. SVM information is not returned for Cluster-scoped interfaces.

```

# The API:
/api/network/ip/interfaces/{uuid}

# The call:
curl -X GET "https://<mgmt-ip>/api/network/ip/interfaces/245979de-59fc-11e8-ba55-005056b4340f" -H "accept: application/hal+json"

```

```

# The response:
{
  "uuid": "245979de-59fc-11e8-ba55-005056b4340f",
  "name": "cluster_mgmt",
  "ip": {
    "address": "10.63.41.6",
    "netmask": "18",
    "family": "ipv4",
  },
  "enabled": true,
  "state": "up",
  "scope": "cluster",
  "ipspace": {
    "uuid": "114ecfb5-59fc-11e8-ba55-005056b4340f",
    "name": "Default",
    "_links": {
      "self": {
        "href": "/api/network/ipspaces/114ecfb5-59fc-11e8-ba55-005056b4340f"
      }
    }
  },
  "services": [
    "management_core",
    "management_autosupport",
    "management_access"
  ],
  "location": {
    "is_home": true,
    "auto_revert": false,
    "failover": "broadcast_domain_only",
    "node": {
      "uuid": "c1db2904-1396-11e9-bb7d-005056acfcb",
      "name": "user-cluster-01-a",
      "_links": {
        "self": {
          "href": "/api/cluster/nodes/c1db2904-1396-11e9-bb7d-005056acfcb"
        }
      }
    },
    "port": {
      "uuid": "c84d5337-1397-11e9-87c2-005056acfcb",
      "name": "e0d",
      "node": {
        "name": "user-cluster-01-a"
      },
    }
  }
}

```

```
  "_links": {
    "self": {
      "href": "/api/network/ethernet/ports/c84d5337-1397-11e9-87c2-
005056acfccb"
    }
  },
  "home_node": {
    "uuid": "c1db2904-1396-11e9-bb7d-005056acfccb",
    "name": "user-cluster-01-a",
    "_links": {
      "self": {
        "href": "/api/cluster/nodes/c1db2904-1396-11e9-bb7d-005056acfccb"
      }
    }
  },
  "home_port": {
    "uuid": "c84d5337-1397-11e9-87c2-005056acfccb",
    "name": "e0d",
    "node": {
      "name": "user-cluster-01-a"
    },
    "_links": {
      "self": {
        "href": "/api/network/ethernet/ports/c84d5337-1397-11e9-87c2-
005056acfccb"
      }
    }
  }
},
"service_policy": {
  "uuid": "9e0f4151-141b-11e9-851e-005056ac1ce0",
  "name": "default-management"
},
"vip": false,
"_links": {
  "self": {
    "href": "/api/network/ip/interfaces/245979de-59fc-11e8-ba55-
005056b4340f"
  }
}
}
```

Retrieving a specific SVM-scoped interface using a filter

The following example shows the response when a specific SVM-scoped interface is requested. The SVM object is only included for SVM-scoped interfaces.

```
# The API:  
/api/network/ip/interfaces  
  
# The call:  
curl -X GET "https://<mgmt-  
ip>/api/network/ip/interfaces?name=lif1&fields=*" -H "accept:  
application/hal+json"  
  
# The response:  
{  
  "records": [  
    {  
      "uuid": "c670707c-5a11-11e8-8fcb-005056b4340f",  
      "name": "lif1",  
      "ip": {  
        "address": "10.10.10.11",  
        "netmask": "24",  
        "family": "ipv4",  
      },  
      "enabled": true,  
      "state": "up",  
      "scope": "svm",  
      "ipspace": {  
        "uuid": "114ecfb5-59fc-11e8-ba55-005056b4340f",  
        "name": "Default",  
        "_links": {  
          "self": {  
            "href": "/api/network/ipspaces/114ecfb5-59fc-11e8-ba55-  
005056b4340f"  
          }  
        }  
      },  
      "svm": {  
        "uuid": "c2134665-5a11-11e8-8fcb-005056b4340f",  
        "name": "user_vs0",  
        "_links": {  
          "self": {  
            "href": "/api/svm/svms/c2134665-5a11-11e8-8fcb-005056b4340f"  
          }  
        }  
      }  
    }  
  ]  
}
```

```

} ,
"services": [
  "data_core",
  "data_nfs",
  "data_cifs",
  "data_flexcache"
],
"location": {
  "is_home": true,
  "auto_revert": false,
  "failover": "broadcast_domain_only",
  "node": {
    "uuid": "c1db2904-1396-11e9-bb7d-005056acfcb",
    "name": "user-cluster-01-a",
    "_links": {
      "self": {
        "href": "/api/cluster/nodes/c1db2904-1396-11e9-bb7d-005056acfcb"
      }
    }
  },
  "port": {
    "uuid": "c84d5337-1397-11e9-87c2-005056acfcb",
    "name": "e0d",
    "node": {
      "name": "user-cluster-01-a"
    },
    "_links": {
      "self": {
        "href": "/api/network/ethernet/ports/c84d5337-1397-11e9-87c2-005056acfcb"
      }
    }
  },
  "home_node": {
    "uuid": "c1db2904-1396-11e9-bb7d-005056acfcb",
    "name": "user-cluster-01-a",
    "_links": {
      "self": {
        "href": "/api/cluster/nodes/c1db2904-1396-11e9-bb7d-005056acfcb"
      }
    }
  },
  "home_port": {
    "uuid": "c84d5337-1397-11e9-87c2-005056acfcb",
    "name": "e0d"
  }
}

```

```

    "name": "e0d",
    "node": {
        "name": "user-cluster-01-a"
    },
    "_links": {
        "self": {
            "href": "/api/network/ethernet/ports/c84d5337-1397-11e9-87c2-
005056acfccb"
        }
    }
},
"service_policy": {
    "uuid": "9e53525f-141b-11e9-851e-005056ac1ce0",
    "name": "default-data-files"
},
"vip": false,
"_links": {
    "self": {
        "href": "/api/network/ip/interfaces/c670707c-5a11-11e8-8fcb-
005056b4340f"
    }
}
}
],
"num_records": 1,
"_links": {
    "self": {
        "href": "/api/network/ip/interfaces?name=lif1&fields=*&filter=location.home_port.name=e0a"
    }
}
}

```

Retrieving specific fields and limiting the output using filters

The following example shows the response when a filter is applied (location.home_port.name=e0a) and only certain fields are requested. Filtered fields are in the output in addition to the default fields and requested fields.

```

# The API:
/api/network/ip/interfaces

# The call:

```

```

curl -X GET "https://<mgmt-
ip>/api/network/ip/interfaces?location.home_port.name=e0a&fields=location.
home_node.name,service_policy.name,ip.address(enabled" -H "accept:
application/hal+json"

# The response:
{
"records": [
{
"uuid": "1d1c9dc8-4f17-11e9-9553-005056ac918a",
"name": "user-cluster-01-a_clus1",
"ip": {
"address": "192.168.170.24"
},
"enabled": true,
"location": {
"home_node": {
"name": "user-cluster-01-a"
},
"home_port": {
"name": "e0a"
}
},
"service_policy": {
"name": "default-cluster"
},
"_links": {
"self": {
"href": "/api/network/ip/interfaces/1d1c9dc8-4f17-11e9-9553-
005056ac918a"
}
}
},
{
"uuid": "d07782c1-4f16-11e9-86e7-005056ace7ee",
"name": "user-cluster-01-b_clus1",
"ip": {
"address": "192.168.170.22"
},
"enabled": true,
"location": {
"home_node": {
"name": "user-cluster-01-b"
},
"home_port": {
"name": "e0a"
}
}
}
]
}
}

```

```

        }
    },
    "service_policy": {
        "name": "default-cluster"
    },
    "_links": {
        "self": {
            "href": "/api/network/ip/interfaces/d07782c1-4f16-11e9-86e7-
005056ace7ee"
        }
    }
},
"num_records": 2,
"_links": {
    "self": {
        "href":-
"/api/network/ip/interfaces?location.home_port.name=e0a&fields=location.ho
me_node.name,service_policy.name,ip.address(enabled"
    }
}
}

```

Creating IP interfaces

You can use the IP interfaces POST API to create IP interfaces as shown in the following examples.

Examples

Creating a Cluster-scoped IP interface using names

The following example shows the record returned after the creation of an IP interface on "e0d".

```

# The API:
/api/network/ip/interfaces

# The call:
curl -X POST "https://<mgmt-
ip>/api/network/ip/interfaces?return_records=true" -H "accept:
application/hal+json" -d '{
    "name": "cluster_mgmt",
    "ip": {
        "address": "10.63.41.6",
        "netmask": "18"
    },
    "enabled": true,
    "scope": "cluster",
    "ipspace": {
        "name": "Default"
    },
    "location": {
        "auto_revert": false
    }
}'

```

```

"failover": "broadcast_domain_only", "home_port": { "name": "e0d", "node": { "name": "user-cluster-01-a" } }, "service_policy": { "name": "default-management" } }

# The response:
{
  "num_records": 1,
  "records": [
    {
      "uuid": "245979de-59fc-11e8-ba55-005056b4340f",
      "name": "cluster_mgmt",
      "ip": {
        "address": "10.63.41.6",
        "netmask": "18"
      },
      "enabled": true,
      "scope": "cluster",
      "ipspace": {
        "name": "Default"
      },
      "location": {
        "auto_revert": false,
        "failover": "broadcast_domain_only",
        "home_port": {
          "name": "e0d",
          "node": {
            "name": "user-cluster-01-a"
          }
        },
        "service_policy": {
          "name": "default-management"
        },
        "_links": {
          "self": {
            "href": "/api/network/ip/interfaces/245979de-59fc-11e8-ba55-005056b4340f"
          }
        }
      }
    }
  ]
}

```

Creating a SVM-scoped IP interface using a mix of parameter types

The following example shows the record returned after the creation of a IP interface by specifying a broadcast domain as the location.

```
# The API:  
/api/network/ip/interfaces  
  
# The call:  
curl -X POST "https://<mgmt-  
ip>/api/network/ip/interfaces?return_records=true" -H "accept:  
application/hal+json" -d '{ "name": "Data1", "ip": { "address":  
"10.234.101.116", "netmask": "255.255.240.0" }, "enabled": true, "scope":  
"svm", "svm": { "uuid": "137f3618-1e89-11e9-803e-005056a7646a" },  
"location": { "auto_revert": true, "broadcast_domain": { "name": "Default"  
} }, "service_policy": { "name": "default-data-files" } }'  
  
# The response:  
{  
  "num_records": 1,  
  "records": [  
    {  
      "uuid": "80d271c9-1f43-11e9-803e-005056a7646a",  
      "name": "Data1",  
      "ip": {  
        "address": "10.234.101.116",  
        "netmask": "20"  
      },  
      "enabled": true,  
      "scope": "svm",  
      "svm": {  
        "uuid": "137f3618-1e89-11e9-803e-005056a7646a",  
        "name": "vs0",  
        "_links": {  
          "self": {  
            "href": "/api/svm/svms/137f3618-1e89-11e9-803e-005056a7646a"  
          }  
        }  
      },  
      "location": {  
        "auto_revert": true  
      },  
      "service_policy": {  
        "name": "default-data-files"  
      },  
    }  
  ]  
}
```

```

  "_links": {
    "self": {
      "href": "/api/network/ip/interfaces/80d271c9-1f43-11e9-803e-
005056a7646a"
    }
  }
}
]
}

```

Creating a Cluster-scoped IP interface without specifying the scope parameter

The following example shows the record returned after creating an IP interface on "e0d" without specifying the scope parameter. The scope is "cluster" if an "svm" is not specified.

```

# The API:
/api/network/ip/interfaces

# The call:
curl -X POST "https://<mgmt-
ip>/api/network/ip/interfaces?return_records=true" -H "accept:
application/hal+json" -d '{ "name": "cluster_mgmt", "ip": { "address":
"10.63.41.6", "netmask": "18" }, "enabled": true, "ipspace": { "name":
"Default" }, "location": { "auto_revert": false, "home_port": { "name":
"e0d", "node": { "name": "user-cluster-01-a" } } }, "service_policy": {
"name": "default-management" } }'

# The response:
{
  "num_records": 1,
  "records": [
    {
      "uuid": "245979de-59fc-11e8-ba55-005056b4340f",
      "name": "cluster_mgmt",
      "ip": {
        "address": "10.63.41.6",
        "netmask": "18"
      },
      "enabled": true,
      "scope": "cluster",
      "ipspace": {
        "name": "Default"
      },
    }
  ]
}

```

```

"location": {
    "auto_revert": false,
    "home_port": {
        "name": "e0d",
        "node": {
            "name": "user-cluster-01-a"
        }
    }
},
"service_policy": {
    "name": "default-management"
},
"_links": {
    "self": {
        "href": "/api/network/ip/interfaces/245979de-59fc-11e8-ba55-005056b4340f"
    }
}
}
]
}

```

Creating an SVM-scoped IP interface without specifying the scope parameter

The following example shows the record returned after creating an IP interface on "e0d" without specifying the scope parameter. The scope is "svm" if the "svm" field is specified.

```

# The API:
/api/network/ip/interfaces

# The call:
curl -X POST "https://<mgmt-
ip>/api/network/ip/interfaces?return_records=true" -H "accept:
application/hal+json" -d '{
    "name": "Data1",
    "ip": {
        "address": "10.234.101.116",
        "netmask": "255.255.240.0"
    },
    "enabled": true,
    "svm": {
        "uuid": "137f3618-1e89-11e9-803e-005056a7646a"
    },
    "location": {
        "auto_revert": true,
        "broadcast_domain": {
            "name": "Default"
        }
    },
    "service_policy": {
        "name": "default-data-files"
    }
}

# The response:
{
    "num_records": 1,
    "records": [

```

```
{
  "uuid": "80d271c9-1f43-11e9-803e-005056a7646a",
  "name": "Data1",
  "ip": {
    "address": "10.234.101.116",
    "netmask": "20"
  },
  "enabled": true,
  "scope": "svm",
  "svm": {
    "uuid": "137f3618-1e89-11e9-803e-005056a7646a",
    "name": "vs0",
    "_links": {
      "self": {
        "href": "/api/svms/137f3618-1e89-11e9-803e-005056a7646a"
      }
    }
  },
  "location": {
    "auto_revert": true
  },
  "service_policy": {
    "name": "default-data-files"
  },
  "_links": {
    "self": {
      "href": "/api/network/ip/interfaces/80d271c9-1f43-11e9-803e-005056a7646a"
    }
  }
}
]
```

Updating IP interfaces

You can use the IP interfaces PATCH API to update the attributes of an IP interface.

Examples

Updating the auto revert flag of an IP interface

The following example shows how the PATCH request changes the auto revert flag to 'false'.

```
# The API:  
/api/network/ip/interfaces/{uuid}  
  
# The call:  
curl -X PATCH "https://<mgmt-ip>/api/network/ip/interfaces/80d271c9-1f43-  
11e9-803e-005056a7646a" -H "accept: application/hal+json" -d '{  
"location": { "auto_revert": "false" } }'  
{  
}
```

Updating the service policy of an IP interface

The following example shows how the PATCH request changes the service policy to 'default-management'.

```
# The API:  
/api/network/ip/interfaces/{uuid}  
  
# The call:  
curl -X PATCH "https://<mgmt-ip>/api/network/ip/interfaces/80d271c9-1f43-  
11e9-803e-005056a7646a" -H "accept: application/hal+json" -d '{  
"service_policy": { "name": "default-management" } }'  
{  
}
```

Deleting IP interfaces

You can use the IP interfaces DELETE API to delete an IP interface in the cluster.

Example

Deleting an IP Interface

The following DELETE request deletes a network IP interface.

```

# The API:
/api/network/ip/interfaces/{uuid}

# The call:
curl -X DELETE "https://<mgmt-ip>/api/network/ip/interfaces/80d271c9-1f43-
11e9-803e-005056a7646a"
{
}

```

Retrieve all IP interface details

GET /network/ip/interfaces

Introduced In: 9.6

Retrieves the details of all IP interfaces.

Related ONTAP Commands

- `network interface show`

Parameters

| Name | Type | In | Required | Description |
|-------------------------|---------|-------|----------|-----------------------------------|
| location.node.name | string | query | False | Filter by location.node.name |
| location.node.uuid | string | query | False | Filter by location.node.uuid |
| location.auto_revert | boolean | query | False | Filter by location.auto_revert |
| location.is_home | boolean | query | False | Filter by location.is_home |
| location.failover | string | query | False | Filter by location.failover |
| location.home_node.name | string | query | False | Filter by location.home_node.name |

| Name | Type | In | Required | Description |
|------------------------------|---------|-------|----------|--|
| location.home_node.uuid | string | query | False | Filter by location.home_node.uuid |
| location.home_port.name | string | query | False | Filter by location.home_port.name |
| location.home_port.node.name | string | query | False | Filter by location.home_port.node.name |
| location.home_port.uuid | string | query | False | Filter by location.home_port.uuid |
| location.port.name | string | query | False | Filter by location.port.name |
| location.port.node.name | string | query | False | Filter by location.port.node.name |
| location.port.uuid | string | query | False | Filter by location.port.uuid |
| enabled | boolean | query | False | Filter by enabled |
| vip | boolean | query | False | Filter by vip |
| scope | string | query | False | Filter by scope |
| services | string | query | False | Filter by services |
| name | string | query | False | Filter by name |
| ipspace.name | string | query | False | Filter by ipspace.name |
| ipspace.uuid | string | query | False | Filter by ipspace.uuid |

| Name | Type | In | Required | Description |
|-------------------------|---------|-------|----------|--|
| metric.status | string | query | False | Filter by metric.status <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.throughput.write | integer | query | False | Filter by metric.throughput.write <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.throughput.read | integer | query | False | Filter by metric.throughput.read <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.throughput.total | integer | query | False | Filter by metric.throughput.total <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.timestamp | string | query | False | Filter by metric.timestamp <ul style="list-style-type: none"> • Introduced in: 9.8 |
| metric.duration | string | query | False | Filter by metric.duration <ul style="list-style-type: none"> • Introduced in: 9.8 |
| uuid | string | query | False | Filter by uuid |
| state | string | query | False | Filter by state |
| ip.address | string | query | False | Filter by ip.address |
| ip.family | string | query | False | Filter by ip.family |
| ip.netmask | string | query | False | Filter by ip.netmask |

| Name | Type | In | Required | Description |
|---------------------------------|---------------|-------|----------|--|
| statistics.timestamp | string | query | False | Filter by statistics.timestamp <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.status | string | query | False | Filter by statistics.status <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.throughput_raw.write | integer | query | False | Filter by statistics.throughput_raw.write <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.throughput_raw.read | integer | query | False | Filter by statistics.throughput_raw.read <ul style="list-style-type: none"> • Introduced in: 9.8 |
| statistics.throughput_raw.total | integer | query | False | Filter by statistics.throughput_raw.total <ul style="list-style-type: none"> • Introduced in: 9.8 |
| svm.uuid | string | query | False | Filter by svm.uuid |
| svm.name | string | query | False | Filter by svm.name |
| service_policy.uuid | string | query | False | Filter by service_policy.uuid |
| service_policy.name | string | query | False | Filter by service_policy.name |
| 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 | <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 |
| 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 • Max value: 120 • Min value: 0 |
| order_by | array[string] | query | False | Order results by specified fields and optional [asc] |

Response

Status: 200, Ok

| Name | Type | Description |
|-------------|---------------------------------------|-------------|
| _links | _links | |
| num_records | integer | |
| records | array[ip_interface] | |

Example response

```
{  
  "_links": {  
    "next": {  
      "href": "/api/resourcelink"  
    },  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "num_records": "1",  
  "records": [  
    {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "ip": {  
        "address": "10.10.10.7",  
        "family": "string",  
        "netmask": "24"  
      },  
      "ipspace": {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        },  
        "name": "exchange",  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
      },  
      "location": {  
        "failover": "string",  
        "home_node": {  
          "_links": {  
            "self": {  
              "href": "/api/resourcelink"  
            }  
          },  
          "name": "node1",  
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        },  
        "home_port": {  
          "_links": {  
            "self": {  
              "href": "/api/resourcelink"  
            }  
          }  
        }  
      }  
    }  
  ]  
}
```

```
        "self": {
            "href": "/api/resourcelink"
        },
        "name": "elb",
        "node": {
            "name": "node1"
        },
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "node": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "name": "node1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "port": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "name": "elb",
        "node": {
            "name": "node1"
        },
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
},
"metric": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "duration": "PT15S",
    "status": "ok",
    "throughput": {
        "read": "200",
        "total": "1000",
        "write": "100"
    },
    "timestamp": "2017-01-25T11:20:13Z"
}
```

```

} ,
  "name": "dataLif1",
  "scope": "string",
  "service_policy": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "default-intercluster",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "services": [
    "data_nfs"
  ],
  "state": "string",
  "statistics": {
    "status": "ok",
    "throughput_raw": {
      "read": "200",
      "total": "1000",
      "write": "100"
    },
    "timestamp": "2017-01-25T11:20:13Z"
  },
  "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 | |

ip_info

IP information

| Name | Type | Description |
|---------|--------|---|
| address | string | IPv4 or IPv6 address |
| family | string | IPv4 or IPv6 |
| netmask | string | Input as netmask length (16) or IPv4 mask (255.255.0.0). For IPv6, you must set the netmask length. The default value is 64. Output is always netmask length. |

ipspace

Either the UUID or name must be supplied on POST for cluster-scoped objects.

| Name | Type | Description |
|--------|--------|--------------|
| _links | _links | |
| name | string | IPspace name |
| uuid | string | IPspace UUID |

broadcast_domain

Broadcast domain UUID along with a readable name.

| Name | Type | Description |
|--------|------------------------|---|
| _links | _links | |
| name | string | Name of the broadcast domain, scoped to its IPspace |
| uuid | string | Broadcast domain UUID |

home_node

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| uuid | string | |

node

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

home_port

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

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| node | node | |
| uuid | string | |

node

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| uuid | string | |

port

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

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| node | node | |
| uuid | string | |

location

Current or home location can be modified. Specifying a port implies a node. Specifying a node allows an appropriate port to be automatically selected. Ports are not valid and not shown for VIP interfaces. For POST, broadcast_domain can be specified alone or with home_node.

| Name | Type | Description |
|-------------|---------------------------|--|
| auto_revert | boolean | |
| failover | string | Defines where an interface may failover. |
| home_node | home_node | |
| home_port | home_port | Port UUID along with readable names. Either the UUID or both names may be supplied on input. |
| is_home | boolean | |
| node | node | |
| port | port | Port UUID along with readable names. Either the UUID or both names may be supplied on input. |

throughput

The rate of throughput bytes per second observed at the interface.

| 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

The most recent sample of I/O metrics for the interface.

| 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: |
| 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 interface. |
| timestamp | string | The timestamp of the performance data. |

service_policy

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| uuid | string | |

throughput_raw

Throughput bytes observed at the interface. This can 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

The real time I/O statistics for the interface.

| Name | Type | Description |
|--------|--------|---|
| 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. |

| Name | Type | Description |
|----------------|----------------|--|
| throughput_raw | throughput_raw | Throughput bytes observed at the interface. This can 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

Applies only to SVM-scoped objects. Either the UUID or name must be supplied on POST.

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

ip_interface

| Name | Type | Description |
|----------|----------|---|
| _links | _links | |
| enabled | boolean | The administrative state of the interface. |
| ip | ip_info | IP information |
| ipspace | ipspace | Either the UUID or name must be supplied on POST for cluster-scoped objects. |
| location | location | Current or home location can be modified. Specifying a port implies a node. Specifying a node allows an appropriate port to be automatically selected. Ports are not valid and not shown for VIP interfaces. For POST, broadcast_domain can be specified alone or with home_node. |

| Name | Type | Description |
|----------------|----------------|---|
| metric | metric | The most recent sample of I/O metrics for the interface. |
| name | string | Interface name |
| scope | string | Set to "svm" for interfaces owned by an SVM. Otherwise, set to "cluster". |
| service_policy | service_policy | |
| services | array[string] | The services associated with the interface. |
| state | string | The operational state of the interface. |
| statistics | statistics | The real time I/O statistics for the interface. |
| svm | svm | Applies only to SVM-scoped objects. Either the UUID or name must be supplied on POST. |
| uuid | string | The UUID that uniquely identifies the interface. |
| vip | boolean | True for a VIP interface, whose location is announced via BGP. |

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 a new cluster-scoped or SVM-scoped interface

POST /network/ip/interfaces

Introduced In: 9.6

Creates a new Cluster-scoped or SVM-scoped interface.

Required properties

- `name` - Name of the interface to create.
- `ip.address` - IP address for the interface.
- `ip.netmask` - IP subnet of the interface.
- `ipspace.name` or `ipspace.uuid`
 - Required for Cluster-scoped interfaces.
 - Optional for SVM-scoped interfaces.
- `svm.name` or `svm.uuid`
 - Required for an SVM-scoped interface.
 - Invalid for a Cluster-scoped interface.
- `location.home_port` or `location.home_node` or `location.broadcast_domain` - One of these properties must be set to a value to define where the interface will be located.

Recommended property values

- `service_policy`
 - for SVM scoped interfaces
 - `default-data-files` for interfaces carrying file-oriented NAS data traffic
 - `default-data-blocks` for interfaces carrying block-oriented SAN data traffic
 - `default-management` for interfaces carrying SVM management requests
 - for Cluster scoped interfaces
 - `default-intercluster` for interfaces carrying cluster peering traffic
 - `default-management` for interfaces carrying system management requests
 - `default-route-announce` for interfaces carrying BGP peer connections

Default property values

If not specified in POST, the following default property values are assigned:

- scope
 - *svm* if *svm* parameter is specified.
 - *cluster* if *svm* parameter is not specified
- *enabled* - *true*
- *location.auto_revert* - *true*
- *service_policy*
 - *default-data-files* if scope is *svm*
 - *default-management* if scope is *cluster* and IPspace is not Cluster
 - *default-cluster* if scope is *svm* and IPspace is Cluster
- *failover* - Selects the least restrictive failover policy supported by all the services in the service policy.

Related ONTAP commands

- `network interface create`

Parameters

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

Request Body

| Name | Type | Description |
|---------|---------|--|
| enabled | boolean | The administrative state of the interface. |
| ip | ip_info | IP information |
| ipspace | ipspace | Either the UUID or name must be supplied on POST for cluster-scoped objects. |

| Name | Type | Description |
|----------------|----------------|---|
| location | location | Current or home location can be modified. Specifying a port implies a node. Specifying a node allows an appropriate port to be automatically selected. Ports are not valid and not shown for VIP interfaces. For POST, broadcast_domain can be specified alone or with home_node. |
| name | string | Interface name |
| scope | string | Set to "svm" for interfaces owned by an SVM. Otherwise, set to "cluster". |
| service_policy | service_policy | |
| services | array[string] | The services associated with the interface. |
| state | string | The operational state of the interface. |
| svm | svm | Applies only to SVM-scoped objects. Either the UUID or name must be supplied on POST. |
| uuid | string | The UUID that uniquely identifies the interface. |
| vip | boolean | True for a VIP interface, whose location is announced via BGP. |

Example request

```
{  
  "ip": {  
    "address": "10.10.10.7",  
    "netmask": "24"  
  },  
  "ipspace": {  
    "name": "exchange",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "location": {  
    "broadcast_domain": {  
      "name": "bd1",  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    },  
    "failover": "string",  
    "home_node": {  
      "name": "node1",  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    },  
    "home_port": {  
      "name": "elb",  
      "node": {  
        "name": "node1"  
      },  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    }  
  },  
  "name": "dataLif1",  
  "scope": "string",  
  "service_policy": {  
    "name": "default-intercluster",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "services": [  
    "data_nfs"  
  ],  
  "state": "string",  
  "svm": {  
    "name": "svm1",  
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
  },  
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
}
```

Response

Status: 201, Created

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|---|
| 1376656 | Cluster interfaces must be in the same subnet. Verify the address and netmask are set to the correct values. |
| 1376963 | Duplicate IP address. |
| 1966138 | The same IP address may not be used for both a mgmt interface and a gateway address. |
| 1966140 | An interface with the same name already exists. |
| 1966987 | The Vserver Broadcast-Domain Home-Node and Home-Port combination is not valid. |
| 1967081 | The specified SVM must exist in the specified IPspace. |
| 1967082 | The specified ipspace.name does not match the IPspace name of ipspace.uuid. |
| 1967102 | POST operation might have left configuration in an inconsistent state. Check the configuration. |
| 1967106 | The specified location.home_port.name does not match the specified port name of location.home_port.uuid. |
| 1967107 | The location.home_port.uuid specified is not valid. |
| 1967108 | The specified location.home_node.name does not match the node name of location.home_node.uuid. |
| 1967109 | The specified location.home_port.node.name does not match the node name of location.home_node.uuid. |
| 1967110 | The specified location.home_port.node.name does not match location.home_node.name. |
| 1967111 | Home node must be specified by at least one location.home_node, location.home_port, or location.broadcast_domain field. |
| 1967112 | The specified location.home_node.name does not match the node name of location.home_port.uuid. |

| Error Code | Description |
|------------|--|
| 1967120 | The specified service_policy.name does not match the specified service policy name of service_policy.uuid. |
| 1967121 | Invalid service_policy.uuid specified. |
| 1967122 | The specified location.broadcast_domain.name does not match the specified broadcast domain name of location.broadcast_domain.uuid. |
| 1967123 | The specified IPspace does not match the IPspace name of location.broadcast_domain.uuid. |
| 1967124 | The location.broadcast_domain.uuid specified is not valid. |
| 1967127 | svm.uuid or svm.name must be provided if scope is "svm". |
| 1967128 | ipspace.uuid or ipspace.name must be provided if scope is "cluster". |
| 1967129 | The specified location.home_port.uuid is not valid. |
| 1967130 | The specified location.home_port.name is not valid. |
| 1967131 | The specified location.home_port.uuid and location.home_port.name are not valid. |
| 1967135 | The specified location.broadcast_domain.uuid is not valid. |
| 1967136 | The specified location.broadcast_domain.name (and ipspace name) is not valid. |
| 1967137 | The specified location.broadcast_domain.uuid and location.broadcast_domain.name (and IPspace name) are not valid. |
| 1967145 | The specified location.failover is not valid. |
| 1967146 | The specified svm.name is not valid. |
| 1967147 | The specified svm.uuid is not valid. |
| 1967153 | No suitable port exists on location.home_node to host the interface. |
| 1967154 | Interfaces cannot be created on ports that are down. If a broadcast domain is specified, ensure that it contains at least one port that is operationally up. |
| 1967381 | Post VIP interfaces requires an effective cluster version of 9.7 or later. |
| 1967382 | VIP interfaces only reside in SVM scope. |
| 1967383 | Neither location.home_port.uuid or location.home_port.name should be set with vip=true. |

| Error Code | Description |
|------------|---|
| 1967384 | Failed to create VIP interface because the home node does not have active BGP sessions to support Virtual IP (VIP) traffic. |
| 1967385 | VIP interfaces with an IPv4 address must use ip.netmask=32. VIP interfaces with an IPv6 address must use ip.netmask=128. |
| 1967387 | The specified IP address is in use by a subnet in this IPspace. |
| 5373966 | An iSCSI interface cannot be created in an SVM configured for NVMe. |
| 53281065 | The service_policy does not exist in the SVM. |

Definitions

See Definitions

href

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

_links

ip_info

IP information

| Name | Type | Description |
|---------|--------|---|
| address | string | IPv4 or IPv6 address |
| netmask | string | Input as netmask length (16) or IPv4 mask (255.255.0.0). For IPv6, you must set the netmask length. The default value is 64. Output is always netmask length. |

ipspace

Either the UUID or name must be supplied on POST for cluster-scoped objects.

| Name | Type | Description |
|------|--------|--------------|
| name | string | IPspace name |
| uuid | string | IPspace UUID |

broadcast_domain

Broadcast domain UUID along with a readable name.

| Name | Type | Description |
|------|--------|---|
| name | string | Name of the broadcast domain, scoped to its IPspace |
| uuid | string | Broadcast domain UUID |

home_node

| Name | Type | Description |
|------|--------|-------------|
| name | string | |

| Name | Type | Description |
|------|--------|-------------|
| uuid | string | |

node

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

home_port

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

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| node | node | |
| uuid | string | |

node

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| uuid | string | |

port

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

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| node | node | |
| uuid | string | |

location

Current or home location can be modified. Specifying a port implies a node. Specifying a node allows an appropriate port to be automatically selected. Ports are not valid and not shown for VIP interfaces. For POST, broadcast_domain can be specified alone or with home_node.

| Name | Type | Description |
|------------------|------------------|---|
| auto_revert | boolean | |
| broadcast_domain | broadcast_domain | Broadcast domain UUID along with a readable name. |

| Name | Type | Description |
|-----------|---------------------------|--|
| failover | string | Defines where an interface may failover. |
| home_node | home_node | |
| home_port | home_port | Port UUID along with readable names. Either the UUID or both names may be supplied on input. |
| is_home | boolean | |

throughput

The rate of throughput bytes per second observed at the interface.

| 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

The most recent sample of I/O metrics for the interface.

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

| Name | Type | Description |
|------------|------------|--|
| status | string | <p>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".</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 | throughput | The rate of throughput bytes per second observed at the interface. |
| timestamp | string | The timestamp of the performance data. |

service_policy

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| uuid | string | |

throughput_raw

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

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

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

statistics

The real time I/O statistics for the interface.

| Name | Type | Description |
|----------------|--------------------------------|---|
| 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_raw | throughput_raw | Throughput bytes observed at the interface. This can 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

Applies only to SVM-scoped objects. Either the UUID or name must be supplied on POST.

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

ip_interface

| Name | Type | Description |
|----------------|--------------------------------|---|
| enabled | boolean | The administrative state of the interface. |
| ip | ip_info | IP information |
| ipspace | ipspace | Either the UUID or name must be supplied on POST for cluster-scoped objects. |
| location | location | Current or home location can be modified. Specifying a port implies a node. Specifying a node allows an appropriate port to be automatically selected. Ports are not valid and not shown for VIP interfaces. For POST, broadcast_domain can be specified alone or with home_node. |
| name | string | Interface name |
| scope | string | Set to "svm" for interfaces owned by an SVM. Otherwise, set to "cluster". |
| service_policy | service_policy | |
| services | array[string] | The services associated with the interface. |
| state | string | The operational state of the interface. |
| svm | svm | Applies only to SVM-scoped objects. Either the UUID or name must be supplied on POST. |

| Name | Type | Description |
|------|---------|--|
| uuid | string | The UUID that uniquely identifies the interface. |
| vip | boolean | True for a VIP interface, whose location is announced via BGP. |

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 IP interface

DELETE /network/ip/interfaces/{uuid}

Introduced In: 9.6

Deletes an IP interface.

Related ONTAP commands

- `network interface delete`

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|-------------------|
| uuid | string | path | True | IP interface UUID |

Response

Status: 200, Ok

Retrieve details for an IP interface

GET /network/ip/interfaces/{uuid}

Introduced In: 9.6

Retrieves details for a specific IP interface.

Related ONTAP commands

- `network interface show`

Parameters

| Name | Type | In | Required | Description |
|--------|---------------|-------|----------|-------------------------------|
| uuid | string | path | True | IP interface UUID |
| fields | array[string] | query | False | Specify the fields to return. |

Response

Status: 200, Ok

| Name | Type | Description |
|---------|-------------------------|--|
| _links | _links | |
| enabled | boolean | The administrative state of the interface. |
| ip | ip_info | IP information |
| ipspace | ipspace | Either the UUID or name must be supplied on POST for cluster-scoped objects. |

| Name | Type | Description |
|----------------|----------------|---|
| location | location | Current or home location can be modified. Specifying a port implies a node. Specifying a node allows an appropriate port to be automatically selected. Ports are not valid and not shown for VIP interfaces. For POST, broadcast_domain can be specified alone or with home_node. |
| metric | metric | The most recent sample of I/O metrics for the interface. |
| name | string | Interface name |
| scope | string | Set to "svm" for interfaces owned by an SVM. Otherwise, set to "cluster". |
| service_policy | service_policy | |
| services | array[string] | The services associated with the interface. |
| state | string | The operational state of the interface. |
| statistics | statistics | The real time I/O statistics for the interface. |
| svm | svm | Applies only to SVM-scoped objects. Either the UUID or name must be supplied on POST. |
| uuid | string | The UUID that uniquely identifies the interface. |
| vip | boolean | True for a VIP interface, whose location is announced via BGP. |

Example response

```
{  
  "_links": {  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "ip": {  
    "address": "10.10.10.7",  
    "family": "string",  
    "netmask": "24"  
  },  
  "ipspace": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    },  
    "name": "exchange",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "location": {  
    "failover": "string",  
    "home_node": {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "name": "node1",  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    },  
    "home_port": {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "name": "elb",  
      "node": {  
        "name": "node1"  
      },  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    },  
    "node": {  
      "name": "node1"  
    }  
  }  
}
```

```
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "node1",
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"port": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "elb",
  "node": {
    "name": "node1"
  },
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"metric": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "duration": "PT15S",
  "status": "ok",
  "throughput": {
    "read": "200",
    "total": "1000",
    "write": "100"
  },
  "timestamp": "2017-01-25T11:20:13Z"
},
"name": "dataLif1",
"scope": "string",
"service_policy": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "default-intercluster",
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
```

```

} ,
"services": [
  "data_nfs"
],
"state": "string",
"statistics": {
  "status": "ok",
  "throughput_raw": {
    "read": "200",
    "total": "1000",
    "write": "100"
  },
  "timestamp": "2017-01-25T11:20:13Z"
},
"svm": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "svml1",
  "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 |
|------|------|-------------|
| self | href | |

ip_info

IP information

| Name | Type | Description |
|---------|--------|---|
| address | string | IPv4 or IPv6 address |
| family | string | IPv4 or IPv6 |
| netmask | string | Input as netmask length (16) or IPv4 mask (255.255.0.0). For IPv6, you must set the netmask length. The default value is 64. Output is always netmask length. |

ipspace

Either the UUID or name must be supplied on POST for cluster-scoped objects.

| Name | Type | Description |
|--------|--------|--------------|
| _links | _links | |
| name | string | IPspace name |
| uuid | string | IPspace UUID |

broadcast_domain

Broadcast domain UUID along with a readable name.

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

| Name | Type | Description |
|------|--------|---|
| name | string | Name of the broadcast domain, scoped to its IPspace |
| uuid | string | Broadcast domain UUID |

home_node

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| uuid | string | |

node

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

home_port

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

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| node | node | |
| uuid | string | |

node

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| uuid | string | |

port

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

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

| Name | Type | Description |
|------|----------------------|-------------|
| name | string | |
| node | node | |
| uuid | string | |

location

Current or home location can be modified. Specifying a port implies a node. Specifying a node allows an appropriate port to be automatically selected. Ports are not valid and not shown for VIP interfaces. For POST, broadcast_domain can be specified alone or with home_node.

| Name | Type | Description |
|-------------|---------------------------|--|
| auto_revert | boolean | |
| failover | string | Defines where an interface may failover. |
| home_node | home_node | |
| home_port | home_port | Port UUID along with readable names. Either the UUID or both names may be supplied on input. |
| is_home | boolean | |
| node | node | |
| port | port | Port UUID along with readable names. Either the UUID or both names may be supplied on input. |

throughput

The rate of throughput bytes per second observed at the interface.

| 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

The most recent sample of I/O metrics for the interface.

| 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: |
| 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 interface. |
| timestamp | string | The timestamp of the performance data. |

service_policy

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| uuid | string | |

throughput_raw

Throughput bytes observed at the interface. This can 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

The real time I/O statistics for the interface.

| Name | Type | Description |
|--------|--------|---|
| 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. |

| Name | Type | Description |
|----------------|----------------|--|
| throughput_raw | throughput_raw | Throughput bytes observed at the interface. This can 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

Applies only to SVM-scoped objects. Either the UUID or name must be supplied on POST.

| 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 IP interface

PATCH /network/ip/interfaces/{uuid}

Introduced In: 9.6

Updates an IP interface.

Related ONTAP commands

- `network interface migrate`
- `network interface modify`
- `network interface rename`
- `network interface revert`

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|-------------------|
| uuid | string | path | True | IP interface UUID |

Request Body

| Name | Type | Description |
|----------------|----------------|---|
| enabled | boolean | The administrative state of the interface. |
| ip | ip_info | IP information |
| location | location | Current or home location can be modified. Specifying a port implies a node. Specifying a node allows an appropriate port to be automatically selected. Ports are not valid and not shown for VIP interfaces. For POST, broadcast_domain can be specified alone or with home_node. |
| name | string | Interface name |
| service_policy | service_policy | |
| services | array[string] | The services associated with the interface. |
| state | string | The operational state of the interface. |
| uuid | string | The UUID that uniquely identifies the interface. |

Example request

```
{  
  "ip": {  
    "address": "10.10.10.7",  
    "netmask": "24"  
  },  
  "location": {  
    "failover": "string",  
    "home_node": {  
      "name": "node1",  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    },  
    "home_port": {  
      "name": "elb",  
      "node": {  
        "name": "node1"  
      },  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    },  
    "node": {  
      "name": "node1",  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    },  
    "port": {  
      "name": "elb",  
      "node": {  
        "name": "node1"  
      },  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    }  
  },  
  "name": "dataLif1",  
  "service_policy": {  
    "name": "default-intercluster",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "services": [  
    "data_nfs"  
  ],  
  "state": "string",  
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",  
  "vip": null  
}
```

Response

Status: 200, Ok

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|---|
| 1376963 | Duplicate IP address. |
| 1966138 | The same IP address may not be used for both a mgmt interface and a gateway address. |
| 1967106 | The specified location.home_port.name does not match the specified port name of location.home_port.uuid. |
| 1967107 | The specified location.home_port.uuid is not valid. |
| 1967111 | A home node must be specified by at least one location.home_node, location.home_port, or location.broadcast_domain field. |
| 1967113 | The specified location.port.name does not match the port name of location.port.uuid. |
| 1967114 | The specified location.port.uuid is not valid. |
| 1967115 | The specified location.node.name does not match the node name of location.node.uuid. |
| 1967116 | The specified location.port.node.name does not match the node name of location.node.uuid. |
| 1967117 | The specified location.port.node.name does not match location.node.name. |
| 1967118 | A node must be specified by at least one location.node or location.port field. |
| 1967119 | The specified location.node.name does not match the node name of location.port.uuid. |
| 1967120 | The specified service_policy.name does not match the specified service policy name of service_policy.uuid. |
| 1967121 | The specified service_policy.uuid is not valid. |

| Error Code | Description |
|------------|---|
| 1967125 | You cannot patch the "location.node" or "location.port" fields to migrate interfaces using the iSCSI data protocol. Instead perform the following PATCH operations on the interface: set the "enabled" field to "false"; change one or more "location.home_port" fields to migrate the interface; and then set the "enabled" field to "true". |
| 1967129 | The specified location.home_port.uuid is not valid. |
| 1967130 | The specified location.home_port.name is not valid. |
| 1967131 | The specified location.home_port.uuid and location.home_port.name are not valid. |
| 1967132 | The specified location.port.uuid is not valid. |
| 1967133 | The specified location.port.name is not valid. |
| 1967134 | The specified location.port.uuid and location.port.name are not valid. |
| 1967138 | Cannot patch port for a VIP interface. The specified parameter location.port.uuid is not valid. |
| 1967139 | Cannot patch port for a VIP interface. The specified parameter location.port.name is not valid. |
| 1967140 | Cannot patch port for a VIP interface. The specified parameters location.port.uuid and location.port.name are not valid. |
| 1967141 | Cannot patch home_port for a VIP interface. The specified parameter location.home_port.uuid is not valid. |
| 1967142 | Cannot patch home_port for a VIP interface. The specified parameter location.home_port.name is not valid. |
| 1967143 | Cannot patch home_port for a VIP interface. The specified parameters location.home_port.uuid and location.home_port.name are not valid. |
| 1967145 | The specified location.failover is not valid. |
| 1967153 | No suitable port exists on location.home_node to host the interface. |
| 1967380 | Cannot patch home_port for a VIP interface. The specified parameter location.home_port.node.name is not valid. Consider using location.home_node.name instead. |
| 1967386 | Cannot patch port for a VIP interface. The specified parameter location.port.node.name is not valid. Consider using location.node.name instead. |

| Error Code | Description |
|------------|---|
| 1967387 | The specified IP address is in use by a subnet in this IPspace. |
| 53281065 | The service_policy does not exist in the SVM. |

Definitions

See Definitions

href

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

_links

ip_info

IP information

| Name | Type | Description |
|---------|--------|---|
| address | string | IPv4 or IPv6 address |
| netmask | string | Input as netmask length (16) or IPv4 mask (255.255.0.0). For IPv6, you must set the netmask length. The default value is 64. Output is always netmask length. |

ipspace

Either the UUID or name must be supplied on POST for cluster-scoped objects.

| Name | Type | Description |
|------|--------|--------------|
| name | string | IPspace name |
| uuid | string | IPspace UUID |

broadcast_domain

Broadcast domain UUID along with a readable name.

| Name | Type | Description |
|------|--------|---|
| name | string | Name of the broadcast domain, scoped to its IPspace |
| uuid | string | Broadcast domain UUID |

home_node

| Name | Type | Description |
|------|--------|-------------|
| name | string | |

| Name | Type | Description |
|------|--------|-------------|
| uuid | string | |

node

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

home_port

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

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| node | node | |
| uuid | string | |

node

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| uuid | string | |

port

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

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| node | node | |
| uuid | string | |

location

Current or home location can be modified. Specifying a port implies a node. Specifying a node allows an appropriate port to be automatically selected. Ports are not valid and not shown for VIP interfaces. For POST, broadcast_domain can be specified alone or with home_node.

| Name | Type | Description |
|-------------|---------|--|
| auto_revert | boolean | |
| failover | string | Defines where an interface may failover. |

| Name | Type | Description |
|-----------|-----------|--|
| home_node | home_node | |
| home_port | home_port | Port UUID along with readable names. Either the UUID or both names may be supplied on input. |
| is_home | boolean | |
| node | node | |
| port | port | Port UUID along with readable names. Either the UUID or both names may be supplied on input. |

throughput

The rate of throughput bytes per second observed at the interface.

| 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

The most recent sample of I/O metrics for the interface.

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

| Name | Type | Description |
|------------|------------|--|
| status | string | <p>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".</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 | throughput | The rate of throughput bytes per second observed at the interface. |
| timestamp | string | The timestamp of the performance data. |

service_policy

| Name | Type | Description |
|------|--------|-------------|
| name | string | |
| uuid | string | |

throughput_raw

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

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

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

statistics

The real time I/O statistics for the interface.

| Name | Type | Description |
|----------------|--------------------------------|---|
| 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_raw | throughput_raw | Throughput bytes observed at the interface. This can 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

Applies only to SVM-scoped objects. Either the UUID or name must be supplied on POST.

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

ip_interface

| Name | Type | Description |
|----------------|--------------------------------|---|
| enabled | boolean | The administrative state of the interface. |
| ip | ip_info | IP information |
| location | location | Current or home location can be modified. Specifying a port implies a node. Specifying a node allows an appropriate port to be automatically selected. Ports are not valid and not shown for VIP interfaces. For POST, broadcast_domain can be specified alone or with home_node. |
| name | string | Interface name |
| service_policy | service_policy | |
| services | array[string] | The services associated with the interface. |
| state | string | The operational state of the interface. |
| uuid | string | The UUID that uniquely identifies the interface. |

Retrieve interface historical performance metrics

GET /network/ip/interfaces/{uuid}/metrics

Introduced In: 9.8

Retrieves historical performance metrics for an interface.

Parameters

| Name | Type | In | Required | Description |
|------------------|---------|-------|----------|-------------------------------------|
| status | string | query | False | Filter by status |
| 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 |
| timestamp | string | query | False | Filter by timestamp |
| duration | string | query | False | Filter by duration |
| uuid | string | path | True | Unique identifier of the interface. |

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

| Name | Type | In | Required | Description |
|--|----------------|---------|----------|--|
| return_timeout | integer | query | False | <p>The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached.</p> <ul style="list-style-type: none"> • Default value: 1 • Max value: 120 • Min value: 0 |
| fields | array[string] | query | False | Specify the fields to return. |
| max_records | integer | query | False | Limit the number of records returned. |
| order_by | array[string] | query | False | Order results by specified fields and optional [asc] |
| desc] direction. Default direction is 'asc' for ascending. | return_records | boolean | query | False |

Response

Status: 200, Ok

| Name | Type | Description |
|-------------|------------------------|-------------------|
| _links | _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",  
      "status": "ok",  
      "throughput": {  
        "read": "200",  
        "total": "1000",  
        "write": "100"  
      },  
      "timestamp": "2017-01-25T11:20:13Z"  
    }  
  ]  
}
```

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 | |

throughput

The rate of throughput bytes per second observed at the interface.

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

records

| 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: |

| Name | Type | Description |
|------------|----------------------------|--|
| status | string | <p>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".</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 | throughput | The rate of throughput bytes per second observed at the interface. |
| 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 |

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

Manage network IP routes

Network IP routes endpoint overview

Overview

This endpoint supports the following operations: GET (collection and instance), POST, and DELETE.

Retrieving network routes

You can use the IP routes GET API to retrieve and display relevant information pertaining to the routes configured in the cluster. The API retrieves the list of all routes configured in the cluster, or a specific route. The fields that are returned in the response will differ with the configuration.

Examples

Retrieving all routes in the cluster

The following output shows the list of all routes configured in a cluster.

```
# The API:
/api/network/ip/routes

# The call:
curl -X GET "https://<mgmt-ip>/api/network/ip/routes?fields=*" -H "accept: application/hal+json"

# The response:
{
  "records": [
    {
      "uuid": "5fdfffb0b-62f8-11e8-853d-005056b4c971",
      "ipspace": {
        "uuid": "84f4beb2-616c-11e8-a4df-005056b4c971",
        "name": "Default",
        "_links": {
          "self": {
            "href": "https://<mgmt-ip>/api/network/ip/routes/5fdfffb0b-62f8-11e8-853d-005056b4c971"
          }
        }
      }
    }
  ]
}
```

```

        "href": "/api/network/ipspaces/84f4beb2-616c-11e8-a4df-
005056b4c971"
    }
}
},
"svm": {
    "uuid": "3243312c-62f8-11e8-853d-005056b4c971",
    "name": "vs1",
    "_links": {
        "self": {
            "href": "/api/svm/svms/3243312c-62f8-11e8-853d-005056b4c971"
        }
    }
},
"scope": "svm",
"destination": {
    "address": "10.4.3.14",
    "netmask": "18",
    "family": "ipv4"
},
"gateway": "10.4.3.1",
"_links": {
    "self": {
        "href": "/api/network/ip/routes/5fdfffb0b-62f8-11e8-853d-
005056b4c971"
    }
}
},
{
    "uuid": "84c128d2-62f9-11e8-853d-005056b4c971",
    "ipspace": {
        "uuid": "cc71aadc-62f7-11e8-853d-005056b4c971",
        "name": "ips1",
        "_links": {
            "self": {
                "href": "/api/network/ipspaces/cc71aadc-62f7-11e8-853d-
005056b4c971"
            }
        }
    }
},
"scope": "cluster",
"destination": {
    "address": "::",
    "netmask": "0",
    "family": "ipv6"
}
}

```

```

"gateway": "fd20:8b1e:b255:814e::1",
"_links": {
  "self": {
    "href": "/api/network/ip/routes/84c128d2-62f9-11e8-853d-
005056b4c971"
  }
},
{
  "uuid": "8cc72bcd-616c-11e8-a4df-005056b4c971",
  "ipspace": {
    "uuid": "84f4beb2-616c-11e8-a4df-005056b4c971",
    "name": "Default",
    "_links": {
      "self": {
        "href": "/api/network/ipspaces/84f4beb2-616c-11e8-a4df-
005056b4c971"
      }
    }
  },
  "scope": "cluster",
  "destination": {
    "address": "0.0.0.0",
    "netmask": "0",
    "family": "ipv4"
  },
  "gateway": "10.224.64.1",
  "_links": {
    "self": {
      "href": "/api/network/ip/routes/8cc72bcd-616c-11e8-a4df-
005056b4c971"
    }
  }
},
{
  "uuid": "d63b6eee-62f9-11e8-853d-005056b4c971",
  "ipspace": {
    "uuid": "84f4beb2-616c-11e8-a4df-005056b4c971",
    "name": "Default",
    "_links": {
      "self": {
        "href": "/api/network/ipspaces/84f4beb2-616c-11e8-a4df-
005056b4c971"
      }
    }
  },
}

```

```

"svm": {
    "uuid": "3243312c-62f8-11e8-853d-005056b4c971",
    "name": "vs1",
    "_links": {
        "self": {
            "href": "/api/svm/svms/3243312c-62f8-11e8-853d-005056b4c971"
        }
    }
},
"scope": "svm",
"destination": {
    "address": "fd20:8b1e:b255:814e::",
    "netmask": "64",
    "family": "ipv6"
},
"gateway": "fd20:8b1e:b255:814e::1",
"_links": {
    "self": {
        "href": "/api/network/ip/routes/d63b6eee-62f9-11e8-853d-005056b4c971"
    }
}
},
"num_records": 4,
"_links": {
    "self": {
        "href": "/api/network/ip/routes?fields=*"
    }
}
}

```

Retrieving a specific Cluster-scoped route

The following output shows the returned response when a specific Cluster-scoped route is requested. The system returns an error if there is no route with the requested UUID. SVM information is not returned for Cluster-scoped routes.

```

# The API:
/api/network/ip/routes/{uuid}

# The call:
curl -X GET "https://<mgmt-ip>/api/network/ip/routes/84c128d2-62f9-11e8-
853d-005056b4c971?fields=*" -H "accept: application/hal+json"

# The response:
{
  "uuid": "84c128d2-62f9-11e8-853d-005056b4c971",
  "ipspace": {
    "uuid": "cc71aadc-62f7-11e8-853d-005056b4c971",
    "name": "ips1",
    "_links": {
      "self": {
        "href": "/api/network/ipspaces/cc71aadc-62f7-11e8-853d-005056b4c971"
      }
    }
  },
  "scope": "cluster",
  "destination": {
    "address": "::",
    "netmask": "0",
    "family": "ipv6"
  },
  "gateway": "fd20:8b1e:b255:814e::1",
  "_links": {
    "self": {
      "href": "/api/network/ip/routes/84c128d2-62f9-11e8-853d-005056b4c971"
    }
  }
}

```

Retrieving a specific SVM-scoped route

The following output shows the returned response when a specific SVM-scoped route is requested. The system returns an error if there is no route with the requested UUID. The SVM object is only included for SVM-scoped routes.

```

# The API:
/api/network/ip/routes/{uuid}

# The call:
curl -X GET "https://<mgmt-ip>/api/network/ip/routes/d63b6eee-62f9-11e8-853d-005056b4c971?fields=*" -H "accept: application/hal+json"

# The response:
{
  "uuid": "d63b6eee-62f9-11e8-853d-005056b4c971",
  "ipspace": {
    "uuid": "84f4beb2-616c-11e8-a4df-005056b4c971",
    "name": "Default",
    "_links": {
      "self": {
        "href": "/api/network/ipspaces/84f4beb2-616c-11e8-a4df-005056b4c971"
      }
    }
  },
  "svm": {
    "uuid": "3243312c-62f8-11e8-853d-005056b4c971",
    "name": "vs1",
    "_links": {
      "self": {
        "href": "/api/svm/svms/3243312c-62f8-11e8-853d-005056b4c971"
      }
    }
  },
  "scope": "svm",
  "destination": {
    "address": "fd20:8b1e:b255:814e::",
    "netmask": "64",
    "family": "ipv6"
  },
  "gateway": "fd20:8b1e:b255:814e::1",
  "_links": {
    "self": {
      "href": "/api/network/ip/routes/d63b6eee-62f9-11e8-853d-005056b4c971"
    }
  }
}

```

Creating network routes

You can use the POST API to create an SVM-scoped route by specifying the associated SVM, or a Cluster-scoped route by specifying the associated IPspace.

Examples

Creating a Cluster-scoped route

IPspace is required to create a Cluster-scoped route. If the IPspace is not specified, the route will be created in the Default IPspace. The default destination will be set to "0.0.0.0/0" for IPv4 gateway addresses or "::/0" for IPv6 gateway addresses.

```
# The API:  
/api/network/ip/routes  
  
# The call:  
curl -X POST "https://<mgmt-ip>/api/network/ip/routes?return_records=true"  
-H "accept: application/json" -d '{ "ipspace": { "name": "ips1" },  
"gateway": "10.10.10.1" }'  
  
# The response:  
{  
  "num_records": 1,  
  "records": [  
    {  
      "uuid": "ae583c9e-9ac7-11e8-8bc9-005056bb531",  
      "ipspace": {  
        "name": "ips1"  
      },  
      "gateway": "10.10.10.1"  
    }  
  ]  
}
```

Creating an SVM-scoped route

To create an SVM-scoped route, the associated SVM can be identified by either its UUID or name.

```

# The API:
/api/network/ip/routes

# The call:
curl -X POST "https://<mgmt-ip>/api/network/ip/routes?return_records=true"
-H "accept: application/json" -d '{ "svm": { "name": "vs0" }, "gateway":
"10.10.10.1" }'

# The response:
{
  "num_records": 1,
  "records": [
    {
      "uuid": "38805a91-9ac9-11e8-8bc9-005056bbd531",
      "svm": {
        "name": "vs0"
      },
      "gateway": "10.10.10.1"
    }
  ]
}

```

Deleting network routes

You can use the DELETE API to delete a specific route identified by its UUID.

Example

Deleting a specific route

```

# The API:
/api/network/ip/routes/{uuid}

# The call:
curl -X DELETE "https://<mgmt-ip>/api/network/ip/routes/38805a91-9ac9-
11e8-8bc9-005056bbd531"

```

Retrieve IP routes

GET /network/ip/routes

Introduced In: 9.6

Retrieves the collection of IP routes.

Related ONTAP commands

- `network route show`

Parameters

| Name | Type | In | Required | Description |
|---------------------|---------------|-------|----------|---------------------------------------|
| scope | string | query | False | Filter by scope |
| ipspace.name | string | query | False | Filter by ipspace.name |
| ipspace.uuid | string | query | False | Filter by ipspace.uuid |
| gateway | string | query | False | Filter by gateway |
| 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 |
| destination.address | string | query | False | Filter by destination.address |
| destination.family | string | query | False | Filter by destination.family |
| destination.netmask | string | query | False | Filter by destination.netmask |
| 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 | <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 |
| 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 • Max value: 120 • Min value: 0 |
| order_by | array[string] | query | False | Order results by specified fields and optional [asc] |

Response

Status: 200, Ok

| Name | Type | Description |
|-------------|--|-------------------|
| _links | _links | |
| error | error | |
| num_records | integer | Number of records |
| records | array[network_route] | |

Example response

```
{  
  "_links": {  
    "next": {  
      "href": "/api/resourcelink"  
    },  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "error": {  
    "arguments": [  
      {  
        "code": "string",  
        "message": "string"  
      }  
    ],  
    "code": "4",  
    "message": "entry doesn't exist",  
    "target": "uuid"  
  },  
  "num_records": "1",  
  "records": [  
    {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "destination": {  
        "address": "10.10.10.7",  
        "family": "string",  
        "netmask": "24"  
      },  
      "gateway": "10.1.1.1",  
      "ipspace": {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        },  
        "name": "exchange",  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
      },  
      "scope": "string",  
      "type": "string"  
    }  
  ]  
}
```

```

"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 | |

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. |

_links

| Name | Type | Description |
|------|------|-------------|
| self | href | |

ip_info

IP information

| Name | Type | Description |
|---------|--------|----------------------|
| address | string | IPv4 or IPv6 address |

| Name | Type | Description |
|---------|--------|---|
| family | string | IPv4 or IPv6 |
| netmask | string | Input as netmask length (16) or IPv4 mask (255.255.0.0). For IPv6, you must set the netmask length. The default value is 64. Output is always netmask length. |

ipspace

Applies to both SVM and cluster-scoped objects. Either the UUID or name may be supplied on input.

| Name | Type | Description |
|--------|------------------------|--------------|
| _links | _links | |
| name | string | IPspace name |
| uuid | string | IPspace UUID |

svm

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

network_route

| Name | Type | Description |
|-------------|-------------------------|---|
| _links | _links | |
| destination | ip_info | IP information |
| gateway | string | The IP address of the gateway router leading to the destination. |
| ipspace | ipspace | Applies to both SVM and cluster-scoped objects. Either the UUID or name may be supplied on input. |

| Name | Type | Description |
|-------|--------|---|
| scope | string | Set to "svm" for interfaces owned by an SVM. Otherwise, set to "cluster". |
| svm | svm | |
| uuid | string | The UUID that uniquely identifies the route. |

Create a cluster-scoped or SVM-scoped static route

POST /network/ip/routes

Introduced In: 9.6

Creates a Cluster-scoped or SVM-scoped static route.

Required properties

- gateway - IP address to route packets to.
- SVM-scoped routes
 - svm.name or svm.uuid - SVM that route is applied to.
- cluster-scoped routes
 - There are no additional required fields for Cluster-scoped routes.

Default property values

If not specified in POST, the following default property values are assigned:

- destination - 0.0.0.0/0 for IPv4 or ::/0 for IPv6.
- ipspace.name
 - *Default* for Cluster-scoped routes.
 - Name of the SVM's IPspace for SVM-scoped routes.

Related ONTAP commands

- `network route create`

Parameters

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

Request Body

| Name | Type | Description |
|-------------|---------|---|
| destination | ip_info | IP information |
| gateway | string | The IP address of the gateway router leading to the destination. |
| ipspace | ipspace | Applies to both SVM and cluster-scoped objects. Either the UUID or name may be supplied on input. |
| svm | svm | |
| uuid | string | The UUID that uniquely identifies the route. |

Example request

```
{
  "destination": {
    "address": "10.10.10.7",
    "netmask": "24"
  },
  "gateway": "10.1.1.1",
  "ipspace": {
    "name": "exchange",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "svm": {
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  },
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
```

Response

Status: 201, Created

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

Example response

```
{  
  "num_records": "1",  
  "records": [  
    {  
      "destination": {  
        "address": "10.10.10.7",  
        "netmask": "24"  
      },  
      "gateway": "10.1.1.1",  
      "ipspace": {  
        "name": "exchange",  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
      },  
      "svm": {  
        "name": "svm1",  
        "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
      },  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    }  
  ]  
}
```

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|-------------------------|
| 1966345 | Duplicate route exists. |

| Error Code | Description |
|------------|--|
| 1967080 | The destination.address is missing. |
| 1967081 | The specified SVM must exist in the specified IPspace. |
| 1967082 | The specified ipspace.uuid and ipspace.name refer to different IPspaces. |
| 1967146 | The specified svm.name is not valid. |
| 2 | The specified svm.uuid is not valid. |

Definitions

See Definitions

href

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

_links

ip_info

IP information

| Name | Type | Description |
|---------|--------|---|
| address | string | IPv4 or IPv6 address |
| netmask | string | Input as netmask length (16) or IPv4 mask (255.255.0.0). For IPv6, you must set the netmask length. The default value is 64. Output is always netmask length. |

ipspace

Applies to both SVM and cluster-scoped objects. Either the UUID or name may be supplied on input.

| Name | Type | Description |
|------|--------|--------------|
| name | string | IPspace name |
| uuid | string | IPspace UUID |

svm

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

network_route

| Name | Type | Description |
|-------------|---------|----------------|
| destination | ip_info | IP information |

| Name | Type | Description |
|---------|---------|---|
| gateway | string | The IP address of the gateway router leading to the destination. |
| ipspace | ipspace | Applies to both SVM and cluster-scoped objects. Either the UUID or name may be supplied on input. |
| svm | svm | |
| uuid | string | The UUID that uniquely identifies the route. |

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 IP route

DELETE /network/ip/routes/{uuid}

Introduced In: 9.6

Deletes a specific IP route.

Related ONTAP commands

- `network route delete`

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|-------------|
| uuid | string | path | True | Route UUID |

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

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 details for an IP route

GET /network/ip/routes/{uuid}

Introduced In: 9.6

Retrieves the details of a specific IP route.

Related ONTAP commands

- `network route show`

Parameters

| Name | Type | In | Required | Description |
|--------|---------------|-------|----------|-------------------------------|
| uuid | string | path | True | Route UUID |
| fields | array[string] | query | False | Specify the fields to return. |

Response

Status: 200, Ok

| Name | Type | Description |
|-------------|-------------------------|---|
| _links | _links | |
| destination | ip_info | IP information |
| gateway | string | The IP address of the gateway router leading to the destination. |
| ipspace | ipspace | Applies to both SVM and cluster-scoped objects. Either the UUID or name may be supplied on input. |
| scope | string | Set to "svm" for interfaces owned by an SVM. Otherwise, set to "cluster". |
| svm | svm | |
| uuid | string | The UUID that uniquely identifies the route. |

Example response

```
{  
  "_links": {  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "destination": {  
    "address": "10.10.10.7",  
    "family": "string",  
    "netmask": "24"  
  },  
  "gateway": "10.1.1.1",  
  "ipspace": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    },  
    "name": "exchange",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "scope": "string",  
  "svm": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    },  
    "name": "svml1",  
    "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 |
|------|------|-------------|
| self | href | |

ip_info

IP information

| Name | Type | Description |
|---------|--------|---|
| address | string | IPv4 or IPv6 address |
| family | string | IPv4 or IPv6 |
| netmask | string | Input as netmask length (16) or IPv4 mask (255.255.0.0). For IPv6, you must set the netmask length. The default value is 64. Output is always netmask length. |

ipspace

Applies to both SVM and cluster-scoped objects. Either the UUID or name may be supplied on input.

| Name | Type | Description |
|--------|--------|--------------|
| _links | _links | |
| name | string | IPspace name |
| uuid | string | IPspace UUID |

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 network IP service policies

Network IP service-policies endpoint overview

Overview

Service policies are named groupings that define what services are supported by an IP interface. The following operations are supported:

- Creation: POST network/ip/service-policies
- Collection Get: GET network/ip/service-policies
- Instance Get: GET network/ip/service-policies/{uuid}
- Instance Patch: PATCH network/ip/service-policies/{uuid}
- Instance Delete: DELETE network/ip/service-policies/{uuid}

Examples

Retrieving all service policies in the cluster

The following output shows the collection of all service policies configured in a 2-node cluster. By default (without 'field='* parameter), only the UUID and name fields are shown for each entry.

```
# The API:
```

```

/api/network/ethernet/ip/service-policies

# The call:
curl -X GET "https://<mgmt-ip>/api/network/ip/service-policies" -H
"accept: application/hal+json"

# The response:
{
  "records": [
    {
      "uuid": "e4e2f193-c1a3-11e8-bb9d-005056bb88c8",
      "name": "net-intercluster",
      "_links": {
        "self": {
          "href": "/api/network/ip/service-policies/e4e2f193-c1a3-11e8-bb9d-005056bb88c8"
        }
      }
    },
    {
      "uuid": "e4e3f6da-c1a3-11e8-bb9d-005056bb88c8",
      "name": "net-route-announce",
      "_links": {
        "self": {
          "href": "/api/network/ip/service-policies/e4e3f6da-c1a3-11e8-bb9d-005056bb88c8"
        }
      }
    },
    {
      "uuid": "e5111111-c1a3-11e8-bb9d-005056bb88c8",
      "name": "vserver-route-announce",
      "_links": {
        "self": {
          "href": "/api/network/ip/service-policies/e5111111-c1a3-11e8-bb9d-005056bb88c8"
        }
      }
    },
    {
      "uuid": "e6111111-c1a3-11e8-bb9d-005056bb88c8",
      "name": "data-route-announce",
      "_links": {
        "self": {
          "href": "/api/network/ip/service-policies/e6111111-c1a3-11e8-bb9d-005056bb88c8"
        }
      }
    }
  ]
}

```

```
        }
    }
}
],
"num_records": 4,
"_links": {
    "self": {
        "href": "/api/network/ip/service-
policies/?return_records=true&return_timeout=15"
    }
}
}
```

Retrieving a specific service policy (scope=svm)

The following output displays the response when a specific "svm" scoped service policy is requested. Among other parameters, the response contains the svm parameters associated with the service policy. The system returns an error when there is no service policy with the requested UUID.

```

# The API:
/api/network/ip/service-policies/{uuid}

# The call:
curl -X GET "http://<mgmt-ip>/api/network/ip/service-policies/dad323ff-
4ce0-11e9-9372-005056bb91a8?fields=*" -H "accept: application/hal+json"

# The response:
{
  "uuid": "dad323ff-4ce0-11e9-9372-005056bb91a8",
  "name": "default-data-files",
  "scope": "svm",
  "svm": {
    "uuid": "d9060680-4ce0-11e9-9372-005056bb91a8",
    "name": "vs0",
    "_links": {
      "self": {
        "href": "/api/svm/svms/d9060680-4ce0-11e9-9372-005056bb91a8"
      }
    }
  },
  "ipspace": {
    "uuid": "45ec2dee-4ce0-11e9-9372-005056bb91a8",
    "name": "Default",
    "_links": {
      "self": {
        "href": "/api/network/ipspaces/45ec2dee-4ce0-11e9-9372-005056bb91a8"
      }
    }
  },
  "services": [
    "data_core",
    "data_nfs",
    "data_cifs",
    "data_flexcache"
  ],
  "_links": {
    "self": {
      "href": "/api/network/ip/service-policies/dad323ff-4ce0-11e9-9372-
005056bb91a8"
    }
  }
}

```

Retrieving a specific service policy (scope=svm) when requesting commonly used fields

The following output displays the response when commonly used fields are requested for a specific "svm" scoped service policy. Among other parameters, the response contains the svm parameters associated with the service policy. The system returns an error when there is no service policy with the requested UUID.

```
# The API:  
/api/network/ip/service-policies/{uuid}  
  
# The call:  
curl -X GET "https://<mgmt-ip>/api/network/ip/service-policies/e0889ce6-  
1e6a-11e9-89d6-005056bbdc04?fields=name,scope,svm.name,ipspace.name" -H  
"accept: application/hal+json"  
  
# The response:  
{  
  "uuid": "e0889ce6-1e6a-11e9-89d6-005056bbdc04",  
  "name": "test_policy",  
  "scope": "svm",  
  "svm": {  
    "name": "vs0"  
  },  
  "ipspace": {  
    "name": "Default"  
  },  
  "_links": {  
    "self": {  
      "href": "/api/network/ip/service-policies/e0889ce6-1e6a-11e9-89d6-  
005056bbdc04"  
    }  
  }  
}
```

Retrieving a specific service policy (scope=cluster)

The following output displays the response when a specific cluster-scoped service policy is requested. The SVM object is not included for cluster-scoped service policies. A service policy with a scope of "cluster" is associated with an IPspace. The system returns an error when there is no service policy with the requested UUID.

```

# The API:
/api/network/ip/service-policies/{uuid}

# The call:
curl -X GET "https://<mgmt-ip>/api/network/ip/service-policies/4c6b72b9-0f6c-11e9-875d-005056bb21b8?fields=*" -H "accept: application/hal+json"

# The response:
{
  "uuid": "4c6b72b9-0f6c-11e9-875d-005056bb21b8",
  "name": "net-intercluster",
  "scope": "cluster",
  "ipspace": {
    "uuid": "4051f13e-0f6c-11e9-875d-005056bb21b8",
    "name": "Default",
    "_links": {
      "self": {
        "href": "/api/network/ipspaces/4051f13e-0f6c-11e9-875d-005056bb21b8"
      }
    }
  },
  "services": [
    "intercluster_core"
  ],
  "_links": {
    "self": {
      "href": "/api/network/ip/service-policies/4c6b72b9-0f6c-11e9-875d-005056bb21b8"
    }
  }
}

```

Retrieving a specific service policy (scope=cluster) when requesting commonly used fields

The following output displays the response when commonly used fields are requested for a specific "cluster" scoped service policy. The SVM object is not included for cluster-scoped service policies. A service policy with a scope of "cluster" is associated with an IPspace. The system returns an error when there is no service policy with the requested UUID.

```

# The API:
/api/network/ip/service-policies/{uuid}

# The call:
curl -X GET "https://<mgmt-ip>/api/network/ip/service-policies/4c6b72b9-0f6c-11e9-875d-005056bb21b8?fields=name,scope,ipspace.name" -H "accept: application/hal+json"

# The response:
{
  "uuid": "4c6b72b9-0f6c-11e9-875d-005056bb21b8",
  "name": "net-intercluster",
  "scope": "cluster",
  "ipspace": {
    "name": "Default"
  },
  "services": [
    "intercluster_core"
  ],
  "_links": {
    "self": {
      "href": "/api/network/ip/service-policies/4c6b72b9-0f6c-11e9-875d-005056bb21b8"
    }
  }
}

```

Creating service policies

You can use this API to create an SVM-scoped service policy by specifying the associated SVM, or a cluster-scoped service policy by specifying the associated IPspace. If the scope is not specified, it is inferred from the presence of the IPspace or SVM. Cluster scoped service policies will operate on the IPspace "Default" unless IPspace is explicitly specified.

Examples

Creating a cluster-scoped service policy

The following output displays the response when creating a service policy with a scope of "cluster" and an IPspace of "Default".

```

# The API:
/api/network/ip/service-policies

# The call:
curl -X POST "https://<mgmt-ip>/api/network/ip/service-
policies?return_records=true" -H "accept: application/json" -d '{
  "name": "new-policy",
  "scope": "cluster",
  "ipspace": {
    "name": "Default"
  },
  "services": [
    "intercluster_core"
  ]
}'

# The response:
{
  "num_records": 1,
  "records": [
    {
      "uuid": "74139267-f1aa-11e9-b5d7-005056a73e2e",
      "name": "new-policy",
      "scope": "cluster",
      "ipspace": {
        "uuid": "ba556295-e912-11e9-a1c8-005056a7080e",
        "name": "Default",
        "_links": {
          "self": {
            "href": "/api/network/ipspaces/ba556295-e912-11e9-a1c8-
005056a7080e"
          }
        }
      },
      "_links": {
        "self": {
          "href": "/api/network/ip/service-policies/74139267-f1aa-11e9-b5d7-
005056a73e2e"
        }
      },
      "services": [
        "intercluster_core"
      ]
    }
  ]
}

```

Creating a cluster-scoped service policy without specifying IPspace

The following output displays the response when creating a service policy with a scope of "cluster" without specifying an IPspace".

```

# The API:
/api/network/ip/service-policies

# The call:
curl -X POST "https://<mgmt-ip>/api/network/ip/service-
policies?return_records=true" -H "accept: application/json" -d '{ "name":'
"new-policy", "scope": "cluster", "services": [ "intercluster_core" ] }'

# The response:
{
  "num_records": 1,
  "records": [
    {
      "uuid": "74139267-f1aa-11e9-b5d7-005056a73e2e",
      "name": "new-policy",
      "scope": "cluster",
      "ipspace": {
        "uuid": "ba556295-e912-11e9-a1c8-005056a7080e",
        "name": "Default",
        "_links": {
          "self": {
            "href": "/api/network/ipspaces/ba556295-e912-11e9-a1c8-
005056a7080e"
          }
        }
      },
      "_links": {
        "self": {
          "href": "/api/network/ip/service-policies/74139267-f1aa-11e9-b5d7-
005056a73e2e"
        }
      },
      "services": [
        "intercluster_core"
      ]
    }
  ]
}

```

Creating a cluster-scoped service policy without specifying scope

The following output displays the response when creating a service policy in the "Default" IPspace without specifying the scope".

```

# The API:
/api/network/ip/service-policies

# The call:
curl -X POST "https://<mgmt-ip>/api/network/ip/service-
policies?return_records=true" -H "accept: application/json" -d '{
  "name": "new-policy2",
  "ipspace.name": "Default",
  "services": [
    "intercluster_core"
  ]
}'

# The response:
{
  "num_records": 1,
  "records": [
    {
      "uuid": "59439267-f1aa-11e9-b5d7-005056a73e2e",
      "name": "new-policy2",
      "scope": "cluster",
      "ipspace": {
        "uuid": "ba556295-e912-11e9-a1c8-005056a7080e",
        "name": "Default",
        "_links": {
          "self": {
            "href": "/api/network/ipspaces/ba556295-e912-11e9-a1c8-
005056a7080e"
          }
        }
      },
      "services": [
        "intercluster_core"
      ],
      "_links": {
        "self": {
          "href": "/api/network/ip/service-policies/74139267-f1aa-11e9-b5d7-
005056a73e2e"
        }
      }
    }
  ]
}

```

Creating an SVM-scoped service policy

The following output displays the response when creating a service policy with a scope of "svm" in the SVM "vs0".

```

# The API:
/api/network/ip/service-policies

# The call:
curl -X POST "https://<mgmt-ip>/api/network/ip/service-
policies?return_records=true" -H "accept: application/json" -d '{ "name": 
"new-policy", "scope": "svm", "svm": { "name": "vs0" }, "services": [ 
"data-nfs", "data-cifs" ] }'

# The response:
{
  "num_records": 1,
  "records": [
    {
      "uuid": "f3901097-f2c4-11e9-b5d7-005056a73e2e",
      "name": "new-policy",
      "scope": "svm",
      "svm": {
        "uuid": "07df9cee-e912-11e9-a13a-005056a73e2e",
        "name": "vs0",
        "_links": {
          "self": {
            "href": "/api/svm/svms/07df9cee-e912-11e9-a13a-005056a73e2e"
          }
        }
      },
      "ipspace": {
        "uuid": "1d3199d2-e906-11e9-a13a-005056a73e2e",
        "name": "Default",
        "_links": {
          "self": {
            "href": "/api/network/ipspaces/1d3199d2-e906-11e9-a13a-
005056a73e2e"
          }
        }
      },
      "services": [
        "data_nfs",
        "data_cifs"
      ],
      "_links": {
        "self": {
          "href": "/api/network/ip/service-policies/f3901097-f2c4-11e9-b5d7-
005056a73e2e"
        }
      }
    }
  ]
}

```

```
    }
}
]
}
```

Creating an SVM-scoped service policy without specifying scope

The following output displays the response when creating a service policy with a SVM of "vs0" without specifying the scope.

```
# The API:
/api/network/ip/service-policies

# The call:
curl -X POST "https://<mgmt-ip>/api/network/ip/service-
policies?return_records=true" -H "accept: application/json" -d '{ "name": 
"new-policy", "svm": { "name": "vs0" }, "services": [ "data-nfs", "data-
cifs" ] }'

# The response:
{
  "num_records": 1,
  "records": [
    {
      "uuid": "f3901097-f2c4-11e9-b5d7-005056a73e2e",
      "name": "new-policy",
      "scope": "svm",
      "svm": {
        "uuid": "07df9cee-e912-11e9-a13a-005056a73e2e",
        "name": "vs0",
        "_links": {
          "self": {
            "href": "/api/svm/svms/07df9cee-e912-11e9-a13a-005056a73e2e"
          }
        }
      },
      "ipspace": {
        "uuid": "1d3199d2-e906-11e9-a13a-005056a73e2e",
        "name": "Default",
        "_links": {
          "self": {
            "href": "/api/network/ipspaces/1d3199d2-e906-11e9-a13a-
005056a73e2e"
          }
        }
      }
    }
  ]
}
```

```

        }
    }
},
"services": [
    "data_nfs",
    "data_cifs"
],
"_links": {
    "self": {
        "href": "/api/network/ip/service-policies/f3901097-f2c4-11e9-b5d7-
005056a73e2e"
    }
}
]
}

```

Updating the name of a service policy

The following example displays the command used to update the name of a service policy scoped to a specific "svm". The system returns an error when there is no service policy associated with the UUID or the service policy cannot be renamed.

```

# The API:
/api/network/ip/service-policies/{uuid}

# The call:
curl -X PATCH "https://<mgmt-ip>/api/network/ip/service-policies/734eaf57-
d2fe-11e9-9284-005056acaad4" -d "{ \"name\": \"new-name\" }" -H "accept:
application/hal+json"

```

Updating the services for a service policy

The following example displays the command used to update the services a service policy contains. The system returns an error when there is no service policy associated with the UUID or the services cannot be applied.

```
# The API:
/api/network/ip/service-policies/{uuid}

# The call:
curl -X PATCH "https://<mgmt-ip>/api/network/ip/service-policies/734eaf57-d2fe-11e9-9284-005056acaad4" -d "{ \"services\": [ \"data-nfs\", \"data-cifs\" ] }" -H "accept: application/hal+json"
```

Deleting a service policy

The following output displays the response for deleting a service policy.

```
# The API:
/api/network/ip/service-policies/{uuid}

# The call:
curl -X DELETE "https://<mgmt-ip>/api/network/ip/service-policies/757ed726-bdc1-11e9-8a92-005056a7bf25" -H "accept: application/hal+json"
```

Retrieve service policies

GET /network/ip/service-policies

Introduced In: 9.6

Retrieves a collection of service policies.

Related ONTAP commands

- `network interface service-policy show`

Parameters

| Name | Type | In | Required | Description |
|----------|--------|-------|----------|--------------------|
| 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 |

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

Response

Status: 200, Ok

| Name | Type | Description |
|-------------|--|-------------|
| _links | _links | |
| num_records | integer | |
| records | array[ip_service_policy] | |

Example response

```
{  
  "_links": {  
    "next": {  
      "href": "/api/resourcelink"  
    },  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "records": [  
    {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "ipspace": {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        },  
        "name": "exchange",  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
      },  
      "name": "default-intercluster",  
      "scope": "string",  
      "services": [  
        "data_nfs"  
      ],  
      "svm": {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        },  
        "name": "svml1",  
        "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 | |

ipspace

| Name | Type | Description |
|--------|------------------------|--------------|
| _links | _links | |
| name | string | IPspace name |
| uuid | string | IPspace UUID |

svm

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

ip_service_policy

| Name | Type | Description |
|---------|-------------------------|-------------|
| _links | _links | |
| ipspace | ipspace | |
| name | string | |

| Name | Type | Description |
|----------|---------------|---|
| scope | string | Set to "svm" for interfaces owned by an SVM. Otherwise, set to "cluster". |
| services | array[string] | |
| svm | svm | |
| uuid | string | |

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 a service policy for network interfaces

POST /network/ip/service-policies

Introduced In: 9.8

Creates a service policy for network interfaces.

Required properties

- `name` - Name of the service policy to create.
- `ipspace.name` or `ipspace.uuid`
 - Required for cluster-scoped service policies.
 - Optional for SVM-scoped service policies.
- `svm.name` or `svm.uuid`

- Required for SVM-scoped service policies.
- Not valid for cluster-scoped service policies.

Default property values

If not specified in POST, the following default property values are assigned:

- scope
 - svm if the svm parameter is specified
 - cluster if the svm parameter is not specified

Parameters

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

Request Body

| Name | Type | Description |
|----------|-------------------------|---|
| ipspace | ipspace | |
| name | string | |
| scope | string | Set to "svm" for interfaces owned by an SVM. Otherwise, set to "cluster". |
| services | array[string] | |
| svm | svm | |
| uuid | string | |

Example request

```
{  
  "ipspace": {  
    "name": "exchange",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "name": "default-intercluster",  
  "scope": "string",  
  "services": [  
    "data_nfs"  
  ],  
  "svm": {  
    "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 |
|------------|---|
| 1967146 | Svm.name does not exist. |
| 1967147 | Svm.uuid does not exist. |
| 53281929 | Service policies cannot combine block and file services. |
| 53281931 | Service policy names cannot start with "default-". |
| 53281933 | Cluster-scoped services cannot be added to a SVM-scoped service policy. |
| 53281934 | SVM-scoped services cannot be added to a Cluster-scoped service policy. |

| Error Code | Description |
|------------|--|
| 53281935 | Scope is set to "svm" and svm.uuid or svm.name have not been specified. |
| 53281936 | The SVM is not in the specified IPspace. |
| 53281937 | Svm.uuid and svm.name are not valid parameters when scope is cluster. |
| 53281938 | Svm.uuid or svm.name specify a vserver that does not exist. |
| 53281939 | One or more of the svm.uuid, svm.name, ipspace.uuid, and ipspace.name have invalid values. |
| 53281940 | SVM or IPspace has not been specified. |
| 53281941 | SVM does not exist. |
| 53281944 | Ipspace.name does not exist. |
| 53281945 | Ipspace.uuid is not an IPspace. |
| 53281946 | Service policy already exists. |

Definitions

See Definitions

href

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

_links

ipspace

| Name | Type | Description |
|------|--------|--------------|
| name | string | IPspace name |
| uuid | string | IPspace UUID |

svm

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

ip_service_policy

| Name | Type | Description |
|----------|---------------|---|
| ipspace | ipspace | |
| name | string | |
| scope | string | Set to "svm" for interfaces owned by an SVM. Otherwise, set to "cluster". |
| services | array[string] | |
| svm | svm | |
| uuid | string | |

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 a service policy for network interfaces

DELETE /network/ip/service-policies/{uuid}

Introduced In: 9.8

Deletes a service policy for network interfaces.

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|----------------------------|
| uuid | string | path | True | UUID of the service policy |

Response

Status: 200, Ok

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|---|
| 53281927 | Service policies owned by the system cannot be deleted. |
| 53281928 | Service policies assigned to LIFs cannot 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 a service policy

GET /network/ip/service-policies/{uuid}

Introduced In: 9.6

Retrieves a specific service policy.

Related ONTAP commands

- `network interface service-policy show`

Parameters

| Name | Type | In | Required | Description |
|----------------|---------------|-------|----------|--|
| uuid | string | path | True | Service policy 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 | <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 |
| 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• Max value: 120• Min value: 0 |

| Name | Type | In | Required | Description |
|----------|---------------|-------|----------|--|
| order_by | array[string] | query | False | Order results by specified fields and optional [asc] |

Response

Status: 200, Ok

| Name | Type | Description |
|----------|-------------------------|---|
| _links | _links | |
| ipspace | ipspace | |
| name | string | |
| scope | string | Set to "svm" for interfaces owned by an SVM. Otherwise, set to "cluster". |
| services | array[string] | |
| svm | svm | |
| uuid | string | |

Example response

```
{  
  "_links": {  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "ipspace": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    },  
    "name": "exchange",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "name": "default-intercluster",  
  "scope": "string",  
  "services": [  
    "data_nfs"  
  ],  
  "svm": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    },  
    "name": "svml1",  
    "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 |
|------|------|-------------|
| self | href | |

ipspace

| Name | Type | Description |
|--------|--------|--------------|
| _links | _links | |
| name | string | IPspace name |
| uuid | string | IPspace UUID |

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 a service policy for network interfaces

PATCH /network/ip/service-policies/{uuid}

Introduced In: 9.8

Updates a service policy for network interfaces.

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|----------------------------|
| uuid | string | path | True | UUID of the service policy |

Request Body

| Name | Type | Description |
|----------|---------------|-------------|
| name | string | |
| services | array[string] | |
| uuid | string | |

Example request

```
{
  "name": "default-intercluster",
  "services": [
    "data_nfs"
  ],
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
```

Response

Status: 200, Ok

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|---|
| 53281929 | Service policies cannot combine block and file services. |
| 53281930 | Service policies maintained by the system cannot be renamed. |
| 53281931 | Service policy names cannot start with "default-". |
| 53281933 | A Cluster-scoped service cannot be added to a SVM-scoped service policy. |
| 53281934 | An SVM-scoped service cannot be added to a Cluster-scoped service policy. |

Definitions

See Definitions

href

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

_links

ipspace

| Name | Type | Description |
|------|--------|--------------|
| name | string | IPspace name |
| uuid | string | IPspace UUID |

svm

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

ip_service_policy

| Name | Type | Description |
|----------|---------------|-------------|
| name | string | |
| services | array[string] | |
| uuid | string | |

error_arguments

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

error

| Name | Type | Description |
|-----------|------------------------|-------------------|
| arguments | array[error_arguments] | Message arguments |

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

Manage network IPspaces

Network ipspaces endpoint overview

Overview

An IPspace is an addressing domain within which each IP address is unique. The same address may appear in a different IPspace, but the matching addresses are considered to be distinct. SVMs and broadcast domains, and therefore IP interfaces and Ethernet ports, are associated with a single IPspace. This endpoint supports the following operations: GET (collection and instance), POST, PATCH, and DELETE.

Retrieving IPspace information

You can use the IPspaces GET API to retrieve all IPspaces configured in the cluster, including built-in and custom IPspaces, and specifically requested IPspaces.

Examples

Retrieving a list of the IPspaces in the cluster

The following example returns the requested list of IPspaces configured in the cluster.

```
# The API:
/api/network/ipspaces

# The call:
curl -X GET "https://<mgmt-ip>/api/network/ipspaces?fields=*" -H "accept: application/hal+json"

# The response:
{
  "records": [
    {
      "uuid": "dcc7e79c-5acc-11e8-b9de-005056b42b32",
      "name": "Default",
      "_links": {
        "self": {
          "href": "/api/network/ipspaces/dcc7e79c-5acc-11e8-b9de-
```

```

005056b42b32"
    }
}
},
{
  "uuid": "dfd3c1b2-5acc-11e8-b9de-005056b42b32",
  "name": "Cluster",
  "_links": {
    "self": {
      "href": "/api/network/ipspacesdfd3c1b2-5acc-11e8-b9de-005056b42b32"
    }
  }
},
{
  "uuid": "dedec1be-5aec-1eee-beee-0eee56be2b3e",
  "name": "Ipspace1",
  "_links": {
    "self": {
      "href": "/api/network/ipspacesdedec1be-5aec-1eee-beee-0eee56be2b3e"
    }
  }
}
],
"num_records": 3,
"_links": {
  "self": {
    "href": "/api/network/ipspaces?fields=*"
  }
}
}

```

Retrieving a specific IPspace in the cluster

The following example returns the specific IPspace requested. The system returns an error if there is no IPspace with the requested UUID.

```
# The API:  
/api/network/ipspaces/{uuid}  
  
# The call:  
curl -X GET "https://<mgmt-ip>/api/network/ipspaces/dfd3c1b2-5acc-11e8-  
b9de-005056b42b32?fields=*" -H "accept: application/hal+json"  
  
# The response:  
{  
  "uuid": "dcc7e79c-5acc-11e8-b9de-005056b42b32",  
  "name": "Default",  
  "_links": {  
    "self": {  
      "href": "/api/network/ipspaces/dcc7e79c-5acc-11e8-b9de-005056b42b32"  
    }  
  }  
}
```

Creating IPspaces

You can use the network IPspaces POST API to create IPspaces.

Example

Creating an IPspace

The following output displays the record returned after the creation of an IPspace with the name "ipspace1".

```

# The API:
/api/network/ipspaces

# The call:
curl -X POST "https://<mgmt-ip>/api/network/ipspaces?return_records=true"
-H "accept: application/hal+json" -d "{  \"name\": \"ipspace2\"}"

# The response:
{
  "num_records": 1,
  "records": [
    {
      "uuid": "4165655e-0528-11e9-bd68-005056bb046a",
      "name": "ipspace2",
      "_links": {
        "self": {
          "href": "/api/network/ipspaces/4165655e-0528-11e9-bd68-
005056bb046a"
        }
      }
    }
  ]
}

```

Updating IPspaces

You can use the IPspaces PATCH API to update the attributes of the IPspace.

Example

Updating the name of an IPspace

The following PATCH request is used to update the name of the IPspace from "ipspace2" to "ipspace20".

```
# The API:  
/api/network/ipspaces/{uuid}  
  
# The call:  
curl -X PATCH "https://<mgmt-ip>/api/network/ipspaces/4165655e-0528-11e9-  
bd68-005056bb046a" -H "accept: application/hal+json" -d "{ \"name\":  
\"ipspace20\"}"
```

Deleting IPspaces

You can use the IPspaces DELETE API to delete an IPspace.

Example

Deleting an IPspace

The following DELETE request is used to delete an IPspace.

```
# The API:  
/api/network/ipspaces/{uuid}  
  
# The call:  
curl -X DELETE "https://<mgmt-ip>/api/network/ipspaces/4165655e-0528-11e9-  
bd68-005056bb046a" -H "accept: application/hal+json" -H "Content-Type:  
application/json"
```

Retrieve IPspaces for a cluster

GET /network/ipspaces

Introduced In: 9.6

Retrieves a collection of IPspaces for the entire cluster.

Related ONTAP commands

- `network ipspace show`

Parameters

| Name | Type | In | Required | Description |
|----------------|---------------|-------|----------|--|
| name | string | query | False | Filter by name |
| 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 | <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 |
| 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 Max value: 120 Min value: 0 |
| order_by | array[string] | query | False | Order results by specified fields and optional [asc] |

Response

Status: 200, Ok

| Name | Type | Description |
|-------------|----------------|-------------|
| _links | _links | |
| num_records | integer | |
| records | array[ipspace] | |

Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": "1",
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "ipspace1",
      "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 | |

ipspace

| Name | Type | Description |
|--------|------------------------|--|
| _links | _links | |
| name | string | IPspace name |
| uuid | string | The UUID that uniquely identifies the IPspace. |

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. |

Create a new domain with unique IP addresses

POST /network/ipspaces

Introduced In: 9.6

Creates a new domain within which IP addresses are unique. SVMs, ports, and networks are scoped to a single IPspace.

Required properties

- name - Name of the IPspace to create.

Related ONTAP commands

- network ipspace create

Parameters

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

Request Body

| Name | Type | Description |
|------|--------|--|
| name | string | IPspace name |
| uuid | string | The UUID that uniquely identifies the IPspace. |

Example request

```
{  
  "name": "ipspace1",  
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
}
```

Response

Status: 201, Created

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|---|
| 1967102 | A POST operation might have left the configuration in an inconsistent state. Check the configuration. |

ONTAP Error Response Codes

| Error Code | Description |
|------------|---|
| 9240591 | The name is not valid. The name is already in use by a cluster node, Vserver, or it is the name of the local cluster. |

Definitions

See Definitions

href

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

_links

ipspace

| Name | Type | Description |
|------|--------|--|
| name | string | IPspace name |
| uuid | string | The UUID that uniquely identifies the IPspace. |

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 IPspace object

DELETE /network/ipspaces/{uuid}

Introduced In: 9.6

Deletes an IPspace object.

Related ONTAP commands

- `network ipspace delete`

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|--------------|
| uuid | string | path | True | IPspace UUID |

Response

```
Status: 200, Ok
```

Retrieve information about an IPspace

GET /network/ipspaces/{uuid}

Introduced In: 9.6

Retrieves information about a specific IPspace.

Related ONTAP commands

- `network ipspace show`

Parameters

| Name | Type | In | Required | Description |
|--------|---------------|-------|----------|-------------------------------|
| uuid | string | path | True | IPspace UUID |
| fields | array[string] | query | False | Specify the fields to return. |

Response

```
Status: 200, Ok
```

| Name | Type | Description |
|--------|------------------------|--|
| _links | _links | |
| name | string | IPspace name |
| uuid | string | The UUID that uniquely identifies the IPspace. |

Example response

```
{  
  "_links": {  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "name": "ipspace1",  
  "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 | |

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 IPspace object

PATCH /network/ipspaces/{uuid}

Introduced In: 9.6

Updates an IPspace object.

Related ONTAP commands

- `network ipspace rename`

Parameters

| Name | Type | In | Required | Description |
|------|--------|------|----------|--------------|
| uuid | string | path | True | IPspace UUID |

Request Body

| Name | Type | Description |
|------|--------|--|
| name | string | IPspace name |
| uuid | string | The UUID that uniquely identifies the IPspace. |

Example request

```
{  
  "name": "ipspace1",  
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
}
```

Response

```
Status: 200, Ok
```

Definitions

See Definitions

href

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

_links

ipspace

| Name | Type | Description |
|------|--------|--|
| name | string | IPspace name |
| uuid | string | The UUID that uniquely identifies the IPspace. |

Copyright information

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—with 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.