Manage network Ethernet ports

ONTAP 9.14.1 REST API reference

NetApp
June 13, 2024

Table of Contents

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, lag (ifgrp), and p-vlan (available in select environments only).

```
# 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"
      }
    }
  },
  {
    "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 for the physical port 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/2d37189f-f70d-
11e8-b145-005056bb5b8e?fields=*" -H "accept: application/hal+json"

# The response:
{
"uuid": "2d37189f-f70d-11e8-b145-005056bb5b8e",
"name": "e0d",
"mac_address": "00:50:56:bb:62:2d",
"type": "physical",
"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,
  "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 for a VLAN port is always set to zero if the state of the port is up. If the state of the port is down, the "speed" field is unset and not reported back.

```
# 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": 0,
"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. The "lag.active_ports" field is set only if the state of the port is up. Also, the "speed" field for a LAG port is always set to zero if the state of the port is up. If the state of the port is down, the "speed" field is unset and not reported back.

```
# 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": 0,
"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": 0,
    "_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": 0,
    "_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.n
ame=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": {
        "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\" } } ]  } }"

# 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"
            }
          }
        ]
      }
    }
  ]
  }
```


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
mtu	integer	query	False	Filter by mtu • Min value: 68

Name	Type	In	Required	Description
mac_address	string	query	False	Filter by mac_address
metric.duration	string	query	False	Filter by metric.duration • Introduced in: 9.8
metric.timestamp	string	query	False	Filter by metric.timestamp • Introduced in: 9.8
metric.throughput.total	integer	query	False	Filter by metric.throughput.total • Introduced in: 9.8
metric.throughput.write	integer	query	False	Filter by metric.throughput.write • Introduced in: 9.8
metric.throughput.read	integer	query	False	Filter by metric.throughput.read • Introduced in: 9.8
metric.status	string	query	False	Filter by metric.status • Introduced in: 9.8
enabled	boolean	query	False	Filter by enabled

Name	Type	In	Required	Description
discovered_devices.capabilities	string	query	False	Filter by discovered_devices.capabilities • Introduced in: 9.11
discovered_devices.protocol	string	query	False	Filter by discovered_devices.protocol • Introduced in: 9.11
discovered_devices.ip_addresses	string	query	False	Filter by discovered_devices.ip_addresses • Introduced in: 9.11
discovered_devices.remote_port	string	query	False	Filter by discovered_devices.remote_port • Introduced in: 9.11
discovered_devices.system_name	string	query	False	Filter by discovered_devices.system_name • Introduced in: 9.11
discovered_devices.name	string	query	False	Filter by discovered_devices.name • Introduced in: 9.11
discovered_devices.version	string	query	False	Filter by discovered_devices.version • Introduced in: 9.11

Name	Type	In	Required	Description
discovered_devices.remaining_hold_time	integer	query	False	Filter by discovered_devices.remaining_hold_time • Introduced in: 9.11
discovered_devices.platform	string	query	False	Filter by discovered_devices.platform • Introduced in: 9.11
discovered_devices.chassis_id	string	query	False	Filter by discovered_devices.chassis_id • Introduced in: 9.11
statistics.throughput_raw.total	integer	query	False	Filter by statistics.throughput_raw.total • Introduced in: 9.8
statistics.throughput_raw.write	integer	query	False	Filter by statistics.throughput_raw.write • Introduced in: 9.8
statistics.throughput_raw.read	integer	query	False	Filter by statistics.throughput_raw.read • Introduced in: 9.8
statistics.timestamp	string	query	False	Filter by statistics.timestamp • Introduced in: 9.8

Name	Type	In	Required	Description
statistics.status	string	query	False	Filter by statistics.status • Introduced in: 9.8
statistics.device.link_down_count_raw	integer	query	False	Filter by statistics.device.link_down_count_raw • Introduced in: 9.8
statistics.device.timestamp	string	query	False	Filter by statistics.device.timestamp • Introduced in: 9.8
statistics.device.transmit_raw.errors	integer	query	False	Filter by statistics.device.transmit_raw.errors • Introduced in: 9.8
statistics.device.transmit_raw.packets	integer	query	False	Filter by statistics.device.transmit_raw.packets • Introduced in: 9.8
statistics.device.transmit_raw.discards	integer	query	False	Filter by statistics.device.transmit_raw.discards • Introduced in: 9.8
statistics.device.receive_raw.errors	integer	query	False	Filter by statistics.device.receive_raw.errors • Introduced in: 9.8

Name	Type	In	Required	Description
statistics.device.receive_raw.packets	integer	query	False	Filter by statistics.device.receive_raw.packets • Introduced in: 9.8
statistics.device.receive_raw.discards	integer	query	False	Filter by statistics.device.receive_raw.discards • Introduced in: 9.8
reachable_broadcast_domains.ipspace.name	string	query	False	Filter by reachable_broadcast_domains.ipspace.name • Introduced in: 9.8
reachable_broadcast_domains.uuid	string	query	False	Filter by reachable_broadcast_domains.uuid • Introduced in: 9.8
reachable_broadcast_domains.name	string	query	False	Filter by reachable_broadcast_domains.name • Introduced in: 9.8
node.name	string	query	False	Filter by node.name
node.uuid	string	query	False	Filter by node.uuid
rdma_protocols	string	query	False	Filter by rdma_protocols • Introduced in: 9.10
uuid	string	query	False	Filter by uuid

Name	Type	In	Required	Description
state	string	query	False	Filter by state
reachability	string	query	False	Filter by reachability • Introduced in: 9.8
type	string	query	False	Filter by type
broadcast_domain.ipspace.name	string	query	False	Filter by broadcast_domain.ipspace.name
broadcast_domain.uuid	string	query	False	Filter by broadcast_domain.uuid
broadcast_domain.name	string	query	False	Filter by broadcast_domain.name
lag.distribution_policy	string	query	False	Filter by lag.distribution_policy
lag.member_ports.uuid	string	query	False	Filter by lag.member_ports.uuid
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.active_ports.uuid	string	query	False	Filter by lag.active_ports.uuid
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

Name	Type	In	Required	Description
lag.mode	string	query	False	Filter by lag.mode
name	string	query	False	Filter by name
vlan.base_port.uuid	string	query	False	Filter by vlan.base_port.uuid
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.tag	integer	query	False	Filter by vlan.tag • Min value: 1 • Max value: 4094
interface_count	integer	query	False	Filter by interface_count • Introduced in: 9.11
speed	integer	query	False	Filter by speed
fields	array[string]	query	False	Specify the fields to return.
max_records	integer	query	False	Limit the number of records returned.
return_records	boolean	query	False	The default is true for GET calls. When set to false, only the number of records is returned. • Default value: 1

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

Response

```
Status: 200, Ok
```

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

Example response


```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": 1,
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "broadcast_domain": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "ipspace": {
          "name": "ipspace1"
        },
        "name": "bd1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      },
      "discovered_devices": [
        {
          "capabilities": [
            "router",
            "switch"
          ],
          "chassis_id": "string",
          "ip_addresses": [
            "192.168.100.24",
            "192.168.100.26"
          ],
          "name": "ETY-R1S4-510Q13.datacenter.example.com",
          "platform": "93180YC-EX",
          "protocol": "cdp",
          "remote_port": "FastEthernet0/12",
```

```
          "system_name": "string",
          "version": "Cisco Nexus Operating System (NX-OS) Software,
Version 8.1"
        }
      ],
      "interface_count": 0,
      "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": "2017-01-25 06:20:13 -0500"
      },
      "mtu": 1500,
      "name": "e1b",
      "node": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "node1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      },
      "rdma_protocols": [
        "string"
      ],
      "reachability": "ok",
      "reachable_broadcast_domains": [
        {
          "_links": {
            "self": {
              "href": "/api/resourcelink"
            }
          },
          "ipspace": {
            "name": "ipspace1"
          },
          "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": "2017-01-25 06:20:13 -0500",
            "transmit_raw": {
              "discards": 100,
              "errors": 200,
              "packets": 500
            }
          },
          "status": "ok",
          "throughput_raw": {
            "read": 200,
            "total": 1000,
            "write": 100
          },
          "timestamp": "2017-01-25 06:20:13 -0500"
        },
        "type": "string",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",
        "vlan": {
          "base_port": {
            "_links": {
              "self": {
                "href": "/api/resourcelink"
              }
            },
            "name": "e1b",
            "node": {
              "name": "node1"
            },
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
          },
          "tag": 100
        }
      }
    ]
  }
```


Error

```
Status: Default, Error
```

Name	Type	Description
error	returned_error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

_links

Name	Type	Description
self	href	

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

discovered_devices

Name	Type	Description
capabilities	array[string]	The list of the capabilities of the discovered device.

Name	Type	Description
chassis_id	string	Identifier associated with this specific discovered device, useful for locating the device in a data center.
ip_addresses	array[string]	The IP addresses on the discovered device.
name	string	Name of the discovered device.
platform	string	Hardware platform of the discovered device.
protocol	string	The protocol used to identify the discovered device. This can have a value of CDP or LLDP.
remaining_hold_time	integer	The number of seconds until the discovered device entry expires and is removed.
remote_port	string	The name of the remote port on the discovered device. The format is dependent on the reporting device.
system_name	string	Additional name used to identify a specific piece of equipment.
version	string	The version of the software running on the discovered device.

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	

Name	Type	Description
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]	Array of ports belonging to the LAG, regardless of their state.
mode	string	Determines how the ports interact with the switch.

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	Peformance 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 interface.
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	Peformance 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.
discovered_devices	array[discovered_devices]	Discovered devices
enabled	boolean	
interface_count	integer	Number of interfaces hosted. This field is only applicable for cluster administrators. No value is returned for SVM administrators. If the node hosting a port is not healthy no value will be returned.
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	Type	Description
name	string	Portname, such as e0a, e1b-100 (VLAN on Ethernet), a0c (LAG/ifgrp), a0d-200 (VLAN on LAG/ifgrp), e0a.pv1 (p-VLAN, in select environments only)
node	node	
rdma_protocols	array[string]	Supported RDMA offload protocols
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. The state is set to 'down' if the operational state of the port is down. The state is set to 'up' if the link state of the port is up and the port is healthy. The state is set to 'up' if the link state of the port is up and configured to ignore health status. The state is 'degraded' if the link state of the port is up, and the port is not healthy.
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

Name	Type	Description
message	string	Message argument

returned_error

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

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.
- `vlan` - This field cannot be specified at the same time as `lag`.
  - `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` - This field cannot be specified at the same time as `vlan`.
  - `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

- `type` - Defines if a VLAN or LAG will be created:
- `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	The default is false. If set to true, the records are returned. • Default value:

Request Body

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.
discovered_devices	array[discovered_devices]	Discovered devices
enabled	boolean	
interface_count	integer	Number of interfaces hosted. This field is only applicable for cluster administrators. No value is returned for SVM administrators. If the node hosting a port is not healthy no value will be returned.
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	Type	Description
name	string	Portname, such as e0a, e1b-100 (VLAN on Ethernet), a0c (LAG/ifgrp), a0d-200 (VLAN on LAG/ifgrp), e0a.pv1 (p-VLAN, in select environments only)
node	node	
rdma_protocols	array[string]	Supported RDMA offload protocols
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. The state is set to 'down' if the operational state of the port is down. The state is set to 'up' if the link state of the port is up and the port is healthy. The state is set to 'up' if the link state of the port is up and configured to ignore health status. The state is 'degraded' if the link state of the port is up, and the port is not healthy.
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	

Example request


```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "broadcast_domain": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "ipspace": {
      "name": "ipspace1"
    },
    "name": "bd1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "discovered_devices": [
    {
      "capabilities": [
        "router",
        "switch"
      ],
      "chassis_id": "string",
      "ip_addresses": [
        "192.168.100.24",
        "192.168.100.26"
      ],
      "name": "ETY-R1S4-510Q13.datacenter.example.com",
      "platform": "93180YC-EX",
      "protocol": "cdp",
      "remote_port": "FastEthernet0/12",
      "system_name": "string",
      "version": "Cisco Nexus Operating System (NX-OS) Software,
Version 8.1"
    }
  ],
  "interface_count": 0,
  "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": "2017-01-25 06:20:13 -0500"
  },
  "mtu": 1500,
  "name": "e1b",
  "node": {
```

```
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "node1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "rdma_protocols": [
    "string"
  ],
  "reachability": "ok",
  "reachable_broadcast_domains": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "ipspace": {
        "name": "ipspace1"
      },
      "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": "2017-01-25 06:20:13 -0500",
      "transmit_raw": {
        "discards": 100,
        "errors": 200,
        "packets": 500
      }
    },
    "status": "ok",
    "throughput_raw": {
      "read": 200,
```

```
      "total": 1000,
      "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
  },
  "type": "string",
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",
  "vlan": {
    "base_port": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "e1b",
      "node": {
        "name": "node1"
      },
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "tag": 100
  }
}
```


Response

```
Status: 201, Created
```

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

Example response


```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": 1,
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "broadcast_domain": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "ipspace": {
          "name": "ipspace1"
        },
        "name": "bd1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      },
      "discovered_devices": [
        {
          "capabilities": [
            "router",
            "switch"
          ],
          "chassis_id": "string",
          "ip_addresses": [
            "192.168.100.24",
            "192.168.100.26"
          ],
          "name": "ETY-R1S4-510Q13.datacenter.example.com",
          "platform": "93180YC-EX",
          "protocol": "cdp",
          "remote_port": "FastEthernet0/12",
```

```
          "system_name": "string",
          "version": "Cisco Nexus Operating System (NX-OS) Software,
Version 8.1"
        }
      ],
      "interface_count": 0,
      "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": "2017-01-25 06:20:13 -0500"
      },
      "mtu": 1500,
      "name": "e1b",
      "node": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "node1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      },
      "rdma_protocols": [
        "string"
      ],
      "reachability": "ok",
      "reachable_broadcast_domains": [
        {
          "_links": {
            "self": {
              "href": "/api/resourcelink"
            }
          },
          "ipspace": {
            "name": "ipspace1"
          },
          "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": "2017-01-25 06:20:13 -0500",
          "transmit_raw": {
            "discards": 100,
            "errors": 200,
            "packets": 500
          }
        },
        "status": "ok",
        "throughput_raw": {
          "read": 200,
          "total": 1000,
          "write": 100
        },
        "timestamp": "2017-01-25 06:20:13 -0500"
      },
      "type": "string",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",
      "vlan": {
        "base_port": {
          "_links": {
            "self": {
              "href": "/api/resourcelink"
            }
          },
          "name": "e1b",
          "node": {
            "name": "node1"
          },
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
        },
        "tag": 100
      }
    }
  ]
}
```


Headers

Name	Description	Type
Location	Useful for tracking the resource location	string

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.
1966466	VLAN ID must be a number from 1 to 4094.
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.
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.

Error Code	Description
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.
1967190	Missing or incomplete VLAN specification.
1967191	Missing or incomplete lag specification.

Also see the table of common errors in the Response body overview section of this documentation.

Name	Type	Description
error	returned_error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

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

discovered_devices

Name	Type	Description
capabilities	array[string]	The list of the capabilities of the discovered device.
chassis_id	string	Identifier associated with this specific discovered device, useful for locating the device in a data center.
ip_addresses	array[string]	The IP addresses on the discovered device.

Name	Type	Description
name	string	Name of the discovered device.
platform	string	Hardware platform of the discovered device.
protocol	string	The protocol used to identify the discovered device. This can have a value of CDP or LLDP.
remaining_hold_time	integer	The number of seconds until the discovered device entry expires and is removed.
remote_port	string	The name of the remote port on the discovered device. The format is dependent on the reporting device.
system_name	string	Additional name used to identify a specific piece of equipment.
version	string	The version of the software running on the discovered device.

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]	Array of ports belonging to the LAG, regardless of their state.
mode	string	Determines how the ports interact with the switch.

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	Peformance metric for write I/O operations.

metric

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

Name	Type	Description
_links	_links	

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

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	Peformance 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.

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 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.
discovered_devices	array[discovered_devices]	Discovered devices
enabled	boolean	
interface_count	integer	Number of interfaces hosted. This field is only applicable for cluster administrators. No value is returned for SVM administrators. If the node hosting a port is not healthy no value will be returned.
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), e0a.pv1 (p-VLAN, in select environments only)
node	node	
rdma_protocols	array[string]	Supported RDMA offload protocols

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. The state is set to 'down' if the operational state of the port is down. The state is set to 'up' if the link state of the port is up and the port is healthy. The state is set to 'up' if the link state of the port is up and configured to ignore health status. The state is 'degraded' if the link state of the port is up, and the port is not healthy.
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	

_links

Name	Type	Description
next	href	
self	href	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

returned_error

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

Delete 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.

Error Code	Description
1966189	Port is the home port or current port of an interface.
1966302	This interface group is hosting VLAN interfaces that must be deleted before running this command.
1967105	Cannot delete a physical port.

Also see the table of common errors in the Response body overview section of this documentation.

Name	Type	Description
error	returned_error	

Example error

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

Definitions

See Definitions

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

returned_error

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

Retrieve 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.
discovered_devices	array[discovered_devices]	Discovered devices
enabled	boolean	
interface_count	integer	Number of interfaces hosted. This field is only applicable for cluster administrators. No value is returned for SVM administrators. If the node hosting a port is not healthy no value will be returned.
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), e0a.pv1 (p-VLAN, in select environments only)
node	node	
rdma_protocols	array[string]	Supported RDMA offload protocols
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.

Name	Type	Description
speed	integer	Link speed in Mbps
state	string	Operational state of the port. The state is set to 'down' if the operational state of the port is down. The state is set to 'up' if the link state of the port is up and the port is healthy. The state is set to 'up' if the link state of the port is up and configured to ignore health status. The state is 'degraded' if the link state of the port is up, and the port is not healthy.
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	

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"
  },
  "discovered_devices": [
    {
      "capabilities": [
        "router",
        "switch"
      ],
      "chassis_id": "string",
      "ip_addresses": [
        "192.168.100.24",
        "192.168.100.26"
      ],
      "name": "ETY-R1S4-510Q13.datacenter.example.com",
      "platform": "93180YC-EX",
      "protocol": "cdp",
      "remote_port": "FastEthernet0/12",
      "system_name": "string",
      "version": "Cisco Nexus Operating System (NX-OS) Software,
Version 8.1"
    }
  ],
  "interface_count": 0,
  "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": "2017-01-25 06:20:13 -0500"
  },
  "mtu": 1500,
  "name": "e1b",
  "node": {
```

```
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "node1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "rdma_protocols": [
    "string"
  ],
  "reachability": "ok",
  "reachable_broadcast_domains": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "ipspace": {
        "name": "ipspace1"
      },
      "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": "2017-01-25 06:20:13 -0500",
      "transmit_raw": {
        "discards": 100,
        "errors": 200,
        "packets": 500
      }
    },
    "status": "ok",
    "throughput_raw": {
      "read": 200,
```

```
      "total": 1000,
      "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
  },
  "type": "string",
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",
  "vlan": {
    "base_port": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "e1b",
      "node": {
        "name": "node1"
      },
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "tag": 100
  }
}
```


Error

```
Status: Default, Error
```

Name	Type	Description
error	returned_error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
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

discovered_devices

Name	Type	Description
capabilities	array[string]	The list of the capabilities of the discovered device.
chassis_id	string	Identifier associated with this specific discovered device, useful for locating the device in a data center.
ip_addresses	array[string]	The IP addresses on the discovered device.

Name	Type	Description
name	string	Name of the discovered device.
platform	string	Hardware platform of the discovered device.
protocol	string	The protocol used to identify the discovered device. This can have a value of CDP or LLDP.
remaining_hold_time	integer	The number of seconds until the discovered device entry expires and is removed.
remote_port	string	The name of the remote port on the discovered device. The format is dependent on the reporting device.
system_name	string	Additional name used to identify a specific piece of equipment.
version	string	The version of the software running on the discovered device.

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]	Array of ports belonging to the LAG, regardless of their state.
mode	string	Determines how the ports interact with the switch.

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	Peformance metric for write I/O operations.

metric

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

Name	Type	Description
_links	_links	

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

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	Peformance 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.

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 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

returned_error

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

Update 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 modify`
- `network port ifgrp add-port`
- `network port ifgrp remove-port`
- `network port reachability repair`

Parameters

Name	Type	In	Required	Description
uuid	string	path	True	Port UUID

Request Body

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.
discovered_devices	array[discovered_devices]	Discovered devices
enabled	boolean	
interface_count	integer	Number of interfaces hosted. This field is only applicable for cluster administrators. No value is returned for SVM administrators. If the node hosting a port is not healthy no value will be returned.
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), e0a.pv1 (p-VLAN, in select environments only)
node	node	
rdma_protocols	array[string]	Supported RDMA offload protocols
reachability	string	Reachability status of the port. Enum value "ok" is the only acceptable value for a PATCH request to repair a port.

Name	Type	Description
reachable_broadcast_domains	array[reachable_broadcast_domains]	Reachable broadcast domains.
speed	integer	Link speed in Mbps
state	string	Operational state of the port. The state is set to 'down' if the operational state of the port is down. The state is set to 'up' if the link state of the port is up and the port is healthy. The state is set to 'up' if the link state of the port is up and configured to ignore health status. The state is 'degraded' if the link state of the port is up, and the port is not healthy.
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	

Example request


```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "broadcast_domain": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "ipspace": {
      "name": "ipspace1"
    },
    "name": "bd1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "discovered_devices": [
    {
      "capabilities": [
        "router",
        "switch"
      ],
      "chassis_id": "string",
      "ip_addresses": [
        "192.168.100.24",
        "192.168.100.26"
      ],
      "name": "ETY-R1S4-510Q13.datacenter.example.com",
      "platform": "93180YC-EX",
      "protocol": "cdp",
      "remote_port": "FastEthernet0/12",
      "system_name": "string",
      "version": "Cisco Nexus Operating System (NX-OS) Software,
Version 8.1"
    }
  ],
  "interface_count": 0,
  "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": "2017-01-25 06:20:13 -0500"
  },
  "mtu": 1500,
  "name": "e1b",
  "node": {
```

```
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "node1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "rdma_protocols": [
    "string"
  ],
  "reachability": "ok",
  "reachable_broadcast_domains": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "ipspace": {
        "name": "ipspace1"
      },
      "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": "2017-01-25 06:20:13 -0500",
      "transmit_raw": {
        "discards": 100,
        "errors": 200,
        "packets": 500
      }
    },
    "status": "ok",
    "throughput_raw": {
      "read": 200,
```

```
      "total": 1000,
      "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
  },
  "type": "string",
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",
  "vlan": {
    "base_port": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "e1b",
      "node": {
        "name": "node1"
      },
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "tag": 100
  }
}
```


Response

```
Status: 200, Ok
```

Error

```
Status: Default
```

ONTAP Error Response Codes

Error Code	Description
1376361	Port is already a member of a LAG.
1376488	Disabling the last operational cluster port on a node is not allowed.
1377563	Port is already a member of a LAG.
1377608	Port cannot be used because it is currently the home port or current port of an interface.

Error Code	Description
1966288	Disabling the cluster ports can only be done on the local node.
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.

Also see the table of common errors in the Response body overview section of this documentation.

Name	Type	Description
error	returned_error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

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

discovered_devices

Name	Type	Description
capabilities	array[string]	The list of the capabilities of the discovered device.
chassis_id	string	Identifier associated with this specific discovered device, useful for locating the device in a data center.
ip_addresses	array[string]	The IP addresses on the discovered device.

Name	Type	Description
name	string	Name of the discovered device.
platform	string	Hardware platform of the discovered device.
protocol	string	The protocol used to identify the discovered device. This can have a value of CDP or LLDP.
remaining_hold_time	integer	The number of seconds until the discovered device entry expires and is removed.
remote_port	string	The name of the remote port on the discovered device. The format is dependent on the reporting device.
system_name	string	Additional name used to identify a specific piece of equipment.
version	string	The version of the software running on the discovered device.

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]	Array of ports belonging to the LAG, regardless of their state.
mode	string	Determines how the ports interact with the switch.

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	Peformance metric for write I/O operations.

metric

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

Name	Type	Description
_links	_links	

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

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	Peformance 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.

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 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.
discovered_devices	array[discovered_devices]	Discovered devices
enabled	boolean	
interface_count	integer	Number of interfaces hosted. This field is only applicable for cluster administrators. No value is returned for SVM administrators. If the node hosting a port is not healthy no value will be returned.
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), e0a.pv1 (p-VLAN, in select environments only)
node	node	
rdma_protocols	array[string]	Supported RDMA offload protocols

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. The state is set to 'down' if the operational state of the port is down. The state is set to 'up' if the link state of the port is up and the port is healthy. The state is set to 'up' if the link state of the port is up and configured to ignore health status. The state is 'degraded' if the link state of the port is up, and the port is not healthy.
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

returned_error

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

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

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
status	string	query	False	Filter by status
throughput.total	integer	query	False	Filter by throughput.total
throughput.write	integer	query	False	Filter by throughput.write
throughput.read	integer	query	False	Filter by throughput.read
timestamp	string	query	False	Filter by timestamp
duration	string	query	False	Filter by duration
uuid	string	path	True	Unique identifier of the port.

Name	Type	In	Required	Description
interval	string	query	False	The time range for the data. Examples can be 1h, 1d, 1m, 1w, 1y. The period for each time range is as follows: • 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	The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached. • Default value: 1 • Max value: 120 • Min value: 0
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"
    }
  },
  "num_records": 1,
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "duration": "PT15S",
      "status": "ok",
      "throughput": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "timestamp": "2017-01-25 06:20:13 -0500",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
  ]
}
```


Error

```
Status: Default, Error
```

Name	Type	Description
error	returned_error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

_links

Name	Type	Description
self	href	

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	Peformance 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	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.
uuid	string	Port UUID

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

returned_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.