# Manage storage qtrees

## REST API reference

NetApp
February 07, 2026

# Table of Contents

# Manage storage qtrees

# Manage storage qtrees

## Overview

A qtree is a logically defined file system that can exist as a special subdirectory of the root directory within a FlexVol volume or a FlexGroup volume.

## Qtree QoS policy

Qtree QoS policy and settings enforce Service Level Objectives (SLOs) on a qtree. SLOs can be set by specifying "qos_policy.max_throughput_iops" and/or "qos_policy.max_throughput_mbps" or "qos_policy.min_throughput_iops" and/or "qos_policy.min_throughput_mbps". Specifying "min_throughput_iops" or "min_throughput_mbps" is only supported on volumes hosted on a node that is flash optimized. A pre-created QoS policy can also be used by specifying "qos_policy.name" or "qos_policy.uuid" properties. Setting or assigning a QoS policy to a qtree is not supported if its containing volume or SVM has a QoS policy attached, or a file or LUN in its containing volume already has a QoS policy attached.

## Performance monitoring

Performance of a qtree can be monitored by observing the `statistics.*` properties. These properties show the performance of the qtree in terms of IOPS and throughput. The `statistics.*` properties denote a real-time monotonically increasing value aggregated across all nodes.

#### Extended performance monitoring

Extended performance monitoring enables the collection of latency statistics and the archival of statistics samples for a qtree. When extended performance monitoring is enabled for a qtree, its performance can be monitored by observing the `metric.*` and `statistics.*` properties. These properties show the performance of the qtree in terms of IOPS, latency and throughput. The `metric.*` properties denote an average whereas `statistics.*` properties denote a real-time monotonically increasing value aggregated across all nodes. Extended performance monitoring can be enabled for a qtree using the `ext_performance_monitoring.enabled` property. A maximum of 50,000 qtrees can have extended performance monitoring enabled on a cluster. If extended performance monitoring is enabled for an existing qtree, the prior `statistics.*` properties showing the IOPS and throughput performance of the qtree will be cleared.

When extended performance monitoring is enabled for a qtree for the first time in an SVM, existing NFS clients might not be accounted for until their NFS shares associated with the SVM are remounted.

## Qtree APIs

The following APIs are used to create, retrieve, modify, and delete qtrees.

– POST /api/storage/qtrees

– GET /api/storage/qtrees

– GET /api/storage/qtrees/{volume-uuid}/{qtree-id}

– PATCH /api/storage/qtrees/{volume-uuid}/{qtree-id}

– DELETE /api/storage/qtrees/{volume-uuid}/{qtree-id}

## Examples

### Creating a qtree inside a volume for an SVM

This API is used to create a qtree inside a volume for an SVM.

The following example shows how to create a qtree in a FlexVol volume with a given security style, user, group, UNIX permissions, an export policy, and a QoS policy.

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

# The call:
curl -X POST 'https://<mgmt-ip>/api/storage/qtrees?return_records=true' -H
'accept: application/hal+json' -d @test_qtree_post.txt
test_qtree_post.txt(body):
{
"svm": {
  "name": "svm1"
},
"volume": {
  "name": "fv"
},
"name": "qt1",
"security_style": "unix",
"user": {
  "name": "unix_user1"
},
"group": {
  "name": "unix_group1"
},
"unix_permissions": 744,
"export_policy": {
    "name": "default"
},
"qos_policy": {
    "max_throughput_iops": 1000
}
}

# The response:
{
  "num_records": 1,
```

```
  "records": [
    {
      "svm": {
        "name": "svm1"
      },
      "volume": {
        "name": "fv"
      },
      "name": "qt1",
      "security_style": "unix",
      "user": {
        "name": "unix_user1"
      },
      "group": {
        "name": "unix_group1"
      },
      "unix_permissions": 744,
      "export_policy": {
        "name": "default"
      },
      "qos_policy": {
        "min_throughput_iops": 0,
        "min_throughput_mbps": 0,
        "max_throughput_iops": 1000,
        "max_throughput_mbps": 0,
        "uuid": "39ac471f-ff35-11e9-b0f9-005056a7ab52",
        "name": "vs0_auto_gen_policy_39a9522f_ff35_11e9_b0f9_005056a7ab52"
      },
      "_links": {
        "self": {
          "href": "/api/storage/qtrees/?volume.name=fv&name=qt1"
        }
      }
    }
  ],
  "job": {
    "uuid": "84edef3c-4f6d-11e9-9a71-005056a7f717",
    "_links": {
      "self": {
        "href": "/api/cluster/jobs/84edef3c-4f6d-11e9-9a71-005056a7f717"
      }
    }
  }
}
```

### Retrieving qtrees

This API is used to retrieve qtrees.

The following example shows how to retrieve qtrees belonging to SVM *svm1* and volume *fv*. The `svm.name` and `volume.name` query parameters are used to find the required qtrees.

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

# The call:
curl -X GET "https://<mgmt-
ip>/api/storage/qtrees/?svm.name=svm1&volume.name=fv" -H 'accept:
application/hal+json'

# The response
{
  "records": [
    {
      "svm": {
        "uuid": "b68f961b-4cee-11e9-930a-005056a7f717",
        "name": "svm1",
        "_links": {
          "self": {
            "href": "/api/svm/svms/b68f961b-4cee-11e9-930a-005056a7f717"
          }
        }
      },
      "volume": {
        "uuid": "cb20da45-4f6b-11e9-9a71-005056a7f717",
        "name": "fv",
        "_links": {
          "self": {
            "href": "/api/storage/volumes/cb20da45-4f6b-11e9-9a71-
005056a7f717"
          }
        }
      },
      "id": 0,
      "name": "",
      "_links": {
        "self": {
          "href": "/api/storage/qtrees/cb20da45-4f6b-11e9-9a71-
005056a7f717/0"
        }
```

```
      }
    },
    {
      "svm": {
        "uuid": "b68f961b-4cee-11e9-930a-005056a7f717",
        "name": "svm1",
        "_links": {
          "self": {
            "href": "/api/svm/svms/b68f961b-4cee-11e9-930a-005056a7f717"
          }
        }
      },
      "volume": {
        "uuid": "cb20da45-4f6b-11e9-9a71-005056a7f717",
        "name": "fv",
        "_links": {
          "self": {
            "href": "/api/storage/volumes/cb20da45-4f6b-11e9-9a71-
005056a7f717"
          }
        }
      },
      "id": 1,
      "name": "qt1",
      "_links": {
        "self": {
          "href": "/api/storage/qtrees/cb20da45-4f6b-11e9-9a71-
005056a7f717/1"
        }
      }
    },
    {
      "svm": {
        "uuid": "b68f961b-4cee-11e9-930a-005056a7f717",
        "name": "svm1",
        "_links": {
          "self": {
            "href": "/api/svm/svms/b68f961b-4cee-11e9-930a-005056a7f717"
          }
        }
      },
      "volume": {
        "uuid": "cb20da45-4f6b-11e9-9a71-005056a7f717",
        "name": "fv",
        "_links": {
          "self": {
```

```
            "href": "/api/storage/volumes/cb20da45-4f6b-11e9-9a71-
005056a7f717"
          }
        }
      },
      "id": 2,
      "name": "qt2",
      "_links": {
        "self": {
          "href": "/api/storage/qtrees/cb20da45-4f6b-11e9-9a71-
005056a7f717/2"
        }
      }
    }
  ],
  "num_records": 3,
  "_links": {
    "self": {
      "href": "/api/storage/qtrees/?svm.name=svm1&volume.name=fv"
    }
  }
}
```

#### Retrieving properties of a specific qtree using a qtree identifier

This API is used to retrieve properties of a specific qtree using qtree.id.

The following example shows how to use the qtree identifier to retrieve all properties of the qtree using the `fields` query parameter.

```
# The API:
GET /api/storage/qtrees/{volume.uuid}/{id}

# The call:
curl -X GET 'https://<mgmt-ip>/api/storage/qtrees/cb20da45-4f6b-11e9-9a71-
005056a7f717/2?fields=*' -H 'accept: application/hal+json'
{
  "svm": {
    "uuid": "b68f961b-4cee-11e9-930a-005056a7f717",
    "name": "svm1",
    "_links": {
      "self": {
        "href": "/api/svm/svms/b68f961b-4cee-11e9-930a-005056a7f717"
```

```
      }
    }
  },
  "volume": {
    "uuid": "cb20da45-4f6b-11e9-9a71-005056a7f717",
    "name": "fv",
    "_links": {
      "self": {
        "href": "/api/storage/volumes/cb20da45-4f6b-11e9-9a71-
005056a7f717"
      }
    }
  },
  "id": 2,
  "name": "qt2",
  "security_style": "unix",
  "user": {
    "name": "unix_user1"
  },
  "group": {
    "name": "unix_group1"
  },
  "unix_permissions": 744,
  "export_policy": {
    "name": "default",
    "id": 12884901889,
    "_links": {
      "self": {
        "href": "/api/protocols/nfs/export-policies/12884901889"
      }
    }
  },
  "qos_policy": {
    "min_throughput_iops": 0,
    "min_throughput_mbps": 0,
    "max_throughput_iops": 1000,
    "max_throughput_mbps": 0,
    "uuid": "39ac471f-ff35-11e9-b0f9-005056a7ab52",
    "name": "vs0_auto_gen_policy_39a9522f_ff35_11e9_b0f9_005056a7ab52",
    "_links": {
      "self": {
        "href": "/api/storage/qos/policies/39ac471f-ff35-11e9-b0f9-
005056a7ab52"
      }
    }
  },
```

```
  "ext_performance_monitoring": {
    "enabled": false
  },
  "statistics": {
    "timestamp": "2019-04-09T05:50:42Z",
    "status": "ok",
    "iops_raw": {
      "read": 0,
      "write": 0,
      "other": 3,
      "total": 3
    },
    "throughput_raw": {
      "read": 0,
      "write": 0,
      "other": 0,
      "total": 0
    }
  },
  "path": "/fv/qt2",
  "nas": {
    "path": "/fv/qt2",
  },
  "_links": {
    "self": {
      "href": "/api/storage/qtrees/cb20da45-4f6b-11e9-9a71-005056a7f717/2"
    }
  }
}
```

#### Retrieving properties of a specific qtree using the qtree name

This API is used to retrieve properties of a specific qtree using "qtree.name". The following example shows how to retrieve all of the properties belonging to qtree "qt2". The `svm.name` and `volume.name` query parameters are used here along with the qtree name.

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

# The call:
curl -X GET 'https://<mgmt-
ip>/api/storage/qtrees/?svm.name=svm1&volume.name=fv&name=qt2&fields=*' -H
'accept: application/hal+json'
```

```
{
  "svm": {
    "uuid": "b68f961b-4cee-11e9-930a-005056a7f717",
    "name": "svm1",
    "_links": {
      "self": {
        "href": "/api/svm/svms/b68f961b-4cee-11e9-930a-005056a7f717"
      }
    }
  },
  "volume": {
    "uuid": "cb20da45-4f6b-11e9-9a71-005056a7f717",
    "name": "fv",
    "_links": {
      "self": {
        "href": "/api/storage/volumes/cb20da45-4f6b-11e9-9a71-
005056a7f717"
      }
    }
  },
  "id": 2,
  "name": "qt2",
  "security_style": "unix",
  "user": {
    "name": "unix_user1"
  },
  "group": {
    "name": "unix_group1"
  },
  "unix_permissions": 744,
  "export_policy": {
    "name": "default",
    "id": 12884901889,
    "_links": {
      "self": {
        "href": "/api/protocols/nfs/export-policies/12884901889"
      }
    }
  },
  "qos_policy": {
    "min_throughput_iops": 0,
    "min_throughput_mbps": 0,
    "max_throughput_iops": 1000,
    "max_throughput_mbps": 0,
    "uuid": "39ac471f-ff35-11e9-b0f9-005056a7ab52",
    "name": "vs0_auto_gen_policy_39a9522f_ff35_11e9_b0f9_005056a7ab52",
```

```
    "_links": {
      "self": {
        "href": "/api/storage/qos/policies/39ac471f-ff35-11e9-b0f9-
005056a7ab52"
      }
    }
  },
  "ext_performance_monitoring": {
    "enabled": false
  },
  "statistics": {
    "timestamp": "2019-04-09T05:50:42Z",
    "status": "ok",
    "iops_raw": {
      "read": 0,
      "write": 0,
      "other": 3,
      "total": 3
    },
    "throughput_raw": {
      "read": 0,
      "write": 0,
      "other": 0,
      "total": 0
    }
  },
  "_links": {
    "self": {
      "href": "/api/storage/qtrees/cb20da45-4f6b-11e9-9a71-005056a7f717/2"
    }
  }
}
```

#### Updating a qtree

This API is used to update a qtree.

The following example shows how to update properties in a qtree.

```
# The API:
PATCH /api/storage/qtrees/{volume.uuid}/{id}

# The call:
curl -X PATCH 'https://<mgmt-ip>/api/storage/qtrees/cb20da45-4f6b-11e9-
9a71-005056a7f717/2' -H 'accept: application/hal+json' -d
'@test_qtree_patch.txt'
test_qtree_patch.txt(body):
{
"security_style": "mixed",
"user": {
  "name": "unix_user1"
},
"group": {
  "name": "unix_group1"
},
"unix_permissions": 777,
"export_policy": {
    "id": "9",
    "name": "exp1"
},
"qos_policy": {
    "uuid": "39ac471f-ff35-11e9-b0f9-005056a7ab53"
}
}
```

#### Renaming a qtree

This API is used to rename a qtree.

The following example below shows how to rename a qtree with a new name.

```
# The API:
PATCH /api/storage/qtrees/{volume.uuid}/{id}

# The call:
curl -X PATCH 'https://<mgmt-ip>/api/storage/qtrees/cb20da45-4f6b-11e9-
9a71-005056a7f717/1' -H 'accept: application/hal+json' -d '{ "name":
"new_qt1" }'
```

#### Deleting a qtree inside a volume of an SVM

This API is used to delete a qtree inside a volume of an SVM.

The following example shows how to delete a qtree.

```
# The API:
DELETE /api/storage/qtrees/{volume.uuid}/{id}

# The call:
curl -X DELETE "https://<mgmt-ip>/api/storage/qtrees/cb20da45-4f6b-11e9-
9a71-005056a7f717/2" -H 'accept: application/hal+json'
```

#### Enabling extended performance monitoring on a qtree using PATCH

This API is used to enable extended performance monitoring on a qtree.

The following example shows how to enable extended performance monitoring on a qtree.

```
# The API:
PATCH /api/storage/qtrees/{volume.uuid}/{id}

# The call:
curl -X PATCH "https://<mgmt-ip>/api/storage/qtrees/cb20da45-4f6b-11e9-
9a71-005056a7f717/1" -H 'accept: application/hal+json' -d '{
"ext_performance_monitoring": { "enabled": "true" }}'
```

#### Disabling extended performance monitoring on a qtree using PATCH

This API is used to disable extended performance monitoring on a qtree.

The following example shows how to disable extended performance monitoring on a qtree.

```
# The API:
PATCH /api/storage/qtrees/{volume.uuid}/{id}

# The call:
curl -X PATCH "https://<mgmt-ip>/api/storage/qtrees/cb20da45-4f6b-11e9-
9a71-005056a7f717/1" -H 'accept: application/hal+json' -d '{
"ext_performance_monitoring": { "enabled": "false" }}'
```

#### Retrieving performance properties for a qtree which has extended performance monitoring enabled

The following example shows how to use a GET request to retrieve performance properties for a qtree which has extended performance monitoring enabled.

```
# The API:
GET /api/storage/qtrees/{volume.uuid}/{id}

# The call:
curl -X GET 'https://<mgmt-ip>/api/storage/qtrees/cb20da45-4f6b-11e9-9a71-
005056a7f717/3?fields=ext_performance_monitoring,statistics,metric' -H
'accept: application/hal+json'
{
  "volume": {
    "uuid": "cb20da45-4f6b-11e9-9a71-005056a7f717",
    "name": "fv",
    "_links": {
      "self": {
        "href": "/api/storage/volumes/cb20da45-4f6b-11e9-9a71-
005056a7f717"
      }
    }
  },
  "id": 3,
  "name": "qt3",
  "ext_performance_monitoring": {
    "enabled": true
  },
  "statistics": {
    "timestamp": "2019-04-09T05:50:42Z",
    "status": "ok",
    "iops_raw": {
      "read": 0,
      "write": 0,
```

```
      "other": 0,
      "total": 0
    },
    "throughput_raw": {
      "read": 0,
      "write": 0,
      "other": 0,
      "total": 0
    }
    "latency_raw": {
      "read": 0,
      "write": 0,
      "other": 0,
      "total": 0
    }
  },
  "metric": {
    "timestamp": "2019-04-09T05:50:00Z",
    "duration": "PT5M",
    "status": "ok",
    "iops": {
      "read": 0,
      "write": 0,
      "other": 0,
      "total": 0
    },
    "throughput": {
      "read": 0,
      "write": 0,
      "other": 0,
      "total": 0
    },
    "latency": {
      "read": 0,
      "write": 0,
      "other": 0,
      "total": 0
    }
  },
  "_links": {
    "self": {
      "href": "/api/storage/qtrees/cb20da45-4f6b-11e9-9a71-005056a7f717/3"
    }
  }
}
```

# Retrieve qtrees

GET `/storage/qtrees`

**Introduced In:** 9.6

Retrieves qtrees configured for all FlexVol volumes or FlexGroup volumes.

Use the `fields` query parameter to retrieve all properties of the qtree. If the `fields` query parameter is not used, then GET returns the qtree `name` and qtree `id` only.

## Expensive properties

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

- `ext_performance_monitoring.enabled`
- `statistics.*`
- `metric.*`

## Related ONTAP commands

- `qtree show`

## Parameters

| Name | Type | In | Required | Description |
|---|---|---|---|---|
| svm.name | string | query | False | Filter by svm.name |
| svm.uuid | string | query | False | Filter by svm.uuid |
| id | integer | query | False | Filter by id<br>• Max value: 4994<br>• Min value: 0 |
| nas.path | string | query | False | Filter by nas.path<br>• Introduced in: 9.9 |
| metric.latency.other | integer | query | False | Filter by metric.latency.other<br>• Introduced in: 9.16 |

| Name | Type | In | Required | Description |
|---|---|---|---|---|
| metric.latency.total | integer | query | False | Filter by metric.latency.total<br>• Introduced in: 9.16 |
| metric.latency.read | integer | query | False | Filter by metric.latency.read<br>• Introduced in: 9.16 |
| metric.latency.write | integer | query | False | Filter by metric.latency.write<br>• Introduced in: 9.16 |
| metric.status | string | query | False | Filter by metric.status<br>• Introduced in: 9.16 |
| metric.duration | string | query | False | Filter by metric.duration<br>• Introduced in: 9.16 |
| metric.timestamp | string | query | False | Filter by metric.timestamp<br>• Introduced in: 9.16 |
| metric.throughput.other | integer | query | False | Filter by metric.throughput.other<br>• Introduced in: 9.16 |
| metric.throughput.total | integer | query | False | Filter by metric.throughput.total<br>• Introduced in: 9.16 |

| Name | Type | In | Required | Description |
|---|---|---|---|---|
| metric.throughput.read | integer | query | False | Filter by metric.throughput.read<br>• Introduced in: 9.16 |
| metric.throughput.write | integer | query | False | Filter by metric.throughput.write<br>• Introduced in: 9.16 |
| metric.iops.other | integer | query | False | Filter by metric.iops.other<br>• Introduced in: 9.16 |
| metric.iops.total | integer | query | False | Filter by metric.iops.total<br>• Introduced in: 9.16 |
| metric.iops.read | integer | query | False | Filter by metric.iops.read<br>• Introduced in: 9.16 |
| metric.iops.write | integer | query | False | Filter by metric.iops.write<br>• Introduced in: 9.16 |
| _tags | string | query | False | Filter by _tags<br>• Introduced in: 9.13 |
| ext_performance_monitoring.enabled | boolean | query | False | Filter by ext_performance_monitoring.enabled<br>• Introduced in: 9.16 |

| Name | Type | In | Required | Description |
|---|---|---|---|---|
| export_policy.id | integer | query | False | Filter by export_policy.id |
| export_policy.name | string | query | False | Filter by export_policy.name |
| qos_policy.min_throughput_mbps | integer | query | False | Filter by qos_policy.min_throughput_mbps<br>• Introduced in: 9.8<br>• Max value: 4194303<br>• Min value: 0 |
| qos_policy.max_throughput_mbps | integer | query | False | Filter by qos_policy.max_throughput_mbps<br>• Introduced in: 9.8<br>• Max value: 4194303<br>• Min value: 0 |
| qos_policy.name | string | query | False | Filter by qos_policy.name<br>• Introduced in: 9.8 |
| qos_policy.uuid | string | query | False | Filter by qos_policy.uuid<br>• Introduced in: 9.8 |
| qos_policy.max_throughput | string | query | False | Filter by qos_policy.max_throughput<br>• Introduced in: 9.17 |

| Name | Type | In | Required | Description |
|---|---|---|---|---|
| qos_policy.max_throughput_iops | integer | query | False | Filter by qos_policy.max_throughput_iops<br>• Introduced in: 9.8<br>• Max value: 2147483647<br>• Min value: 0 |
| qos_policy.min_throughput_iops | integer | query | False | Filter by qos_policy.min_throughput_iops<br>• Introduced in: 9.8<br>• Max value: 2147483647<br>• Min value: 0 |
| qos_policy.min_throughput | string | query | False | Filter by qos_policy.min_throughput<br>• Introduced in: 9.17 |
| user.id | string | query | False | Filter by user.id<br>• Introduced in: 9.9 |
| user.name | string | query | False | Filter by user.name<br>• Introduced in: 9.9 |
| statistics.throughput_raw.other | integer | query | False | Filter by statistics.throughput_raw.other<br>• Introduced in: 9.8 |

| Name | Type | In | Required | Description |
|---|---|---|---|---|
| statistics.throughput_raw.total | integer | query | False | Filter by statistics.throughput_raw.total<br>• Introduced in: 9.8 |
| statistics.throughput_raw.read | integer | query | False | Filter by statistics.throughput_raw.read<br>• Introduced in: 9.8 |
| statistics.throughput_raw.write | integer | query | False | Filter by statistics.throughput_raw.write<br>• Introduced in: 9.8 |
| statistics.iops_raw.other | integer | query | False | Filter by statistics.iops_raw.other<br>• Introduced in: 9.8 |
| statistics.iops_raw.total | integer | query | False | Filter by statistics.iops_raw.total<br>• Introduced in: 9.8 |
| statistics.iops_raw.read | integer | query | False | Filter by statistics.iops_raw.read<br>• Introduced in: 9.8 |
| statistics.iops_raw.write | integer | query | False | Filter by statistics.iops_raw.write<br>• Introduced in: 9.8 |

| Name | Type | In | Required | Description |
|---|---|---|---|---|
| statistics.latency_raw.other | integer | query | False | Filter by statistics.latency_raw.other<br>• Introduced in: 9.16 |
| statistics.latency_raw.total | integer | query | False | Filter by statistics.latency_raw.total<br>• Introduced in: 9.16 |
| statistics.latency_raw.read | integer | query | False | Filter by statistics.latency_raw.read<br>• Introduced in: 9.16 |
| statistics.latency_raw.write | integer | query | False | Filter by statistics.latency_raw.write<br>• Introduced in: 9.16 |
| statistics.timestamp | string | query | False | Filter by statistics.timestamp<br>• Introduced in: 9.8 |
| statistics.status | string | query | False | Filter by statistics.status<br>• Introduced in: 9.8 |
| group.id | string | query | False | Filter by group.id<br>• Introduced in: 9.9 |
| group.name | string | query | False | Filter by group.name<br>• Introduced in: 9.9 |

| Name | Type | In | Required | Description |
|---|---|---|---|---|
| unix_permissions | integer | query | False | Filter by unix_permissions |
| security_style | string | query | False | Filter by security_style |
| volume.uuid | string | query | False | Filter by volume.uuid |
| volume.name | string | query | False | Filter by volume.name |
| path | string | query | False | Filter by path |
| name | string | query | False | Filter by name |
| fields | array[string] | query | False | Specify the fields to return. |
| max_records | integer | query | False | Limit the number of records returned. |
| return_records | boolean | query | False | The default is true for GET calls. When set to false, only the number of records is returned.<br>• Default value: 1 |
| return_timeout | integer | query | False | The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached.<br>• Max value: 120<br>• Min value: 0<br>• Default value: 15 |

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

## Response

```
Status: 200, Ok
```

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

**Example response**


```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": 1,
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "_tags": [
        "team:csi",
        "environment:test"
      ],
      "export_policy": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "id": 100,
        "name": "default"
      },
      "group": {
        "id": "20001",
        "name": "unix_group1"
      },
      "id": 1,
      "metric": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "duration": "PT5M",
        "iops": {
          "read": 200,
```

```
        "total": 1000,
        "write": 100
      },
      "latency": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "status": "ok",
      "throughput": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "timestamp": "2017-01-25 06:20:13 -0500"
    },
    "name": "string",
    "nas": {
      "path": "/volume3/qtree1"
    },
    "path": "/volume3/qtree1",
    "qos_policy": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "max_throughput": [
        "900KB/s",
        "500MB/s",
        "120GB/s",
        "5000IOPS",
        "5000IOPS,500KB/s",
        "2500IOPS,100MB/s",
        "1000IOPS,25MB/s"
      ],
      "max_throughput_iops": 10000,
      "max_throughput_mbps": 500,
      "min_throughput": [
        "900KB/s",
        "500MB/s",
        "120GB/s",
        "5000IOPS",
        "5000IOPS,500KB/s",
        "2500IOPS,100MB/s",
        "1000IOPS,25MB/s"
```

```
      ],
      "min_throughput_iops": 2000,
      "min_throughput_mbps": 500,
      "name": "performance",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "security_style": "string",
    "statistics": {
      "iops_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "latency_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "status": "ok",
      "throughput_raw": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "timestamp": "2017-01-25 06:20:13 -0500"
    },
    "svm": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "svm1",
      "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
    },
    "unix_permissions": 493,
    "user": {
      "id": "10001",
      "name": "unix_user1"
    },
    "volume": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
```

```
        "name": "volume1",
        "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
      }
    }
  ]
}
```


## Error

```
Status: Default
```

ONTAP Error Response Codes

| Error Code | Description |
|---|---|
| 918235 | A volume with UUID was not found. |
| 2621462 | The specified SVM does not exist. |
| 5242889 | Failed to get the qtree from volume. |
| 5242956 | Failed to obtain qtree. |
| 5242965 | Invalid qtree path. The volume name component of the qtree path, must be the same as the volume specified with the parameter. |

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 |
|---|---|---|
| next | href | |
| self | href | |

_links

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

export_policy

Export Policy

| Name | Type | Description |
|---|---|---|
| _links | _links | |
| id | integer | |
| name | string | |

ext_performance_monitoring

| Name | Type | Description |
|---|---|---|
| enabled | boolean | Specifies whether extended performance monitoring is enabled for the qtree. |

group

The user set as owner of the qtree.

| Name | Type | Description |
|---|---|---|
| id | string | The numeric ID of the group that owns the qtree. Valid in POST or PATCH. |

| Name | Type | Description |
|---|---|---|
| name | string | Alphanumeric group name of group that owns the qtree. Valid in POST or PATCH. |

iops

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

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

latency

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

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

throughput

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

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

metric

Performance numbers, such as IOPS latency and throughput.

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

| 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 storage object. |
| timestamp | string | The timestamp of the performance data. |

nas

| Name | Type | Description |
|---|---|---|
| path | string | Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH. |

qos_policy

When "min_throughput_iops", "min_throughput_mbps", "min_throughput", "max_throughput_iops", "max_throughput_mbps" or "max_throughput" attributes are specified, the storage object is assigned to an auto-generated QoS policy group. If the attributes are later modified, the auto-generated QoS policy-group attributes are modified. Attributes can be removed by specifying "0" and policy group by specifying "none". Upon deletion of the storage object or if the attributes are removed, then the QoS policy-group is also removed.

| Name | Type | Description |
| --- | --- | --- |
| _links | _links | |
| max_throughput | string | Specifies the maximum throughput in Kilobytes per sec, Megabytes per sec or Gigabytes per sec along with or without IOPS. 0 means none. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when either max_throughput_mbps or max_throughput_iops are set during POST or PATCH. During GET, the returned value is rounded to the largest unit with a value greater than 1. |
| max_throughput_iops | integer | Specifies the maximum throughput in IOPS, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when max_throughput is set during POST or PATCH. |
| max_throughput_mbps | integer | Specifies the maximum throughput in Megabytes per sec, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when max_throughput is set during POST or PATCH. |

| Name | Type | Description |
| --- | --- | --- |
| min_throughput | string | Specifies the minimum throughput in Kilobytes per sec, Megabytes per sec or Gigabytes per sec along with or without IOPS. 0 means none.Setting "min_throughput" is supported on AFF platforms only, unless FabricPool tiering policies are set. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when either min_throughput_mbps or min_throughput_iops are set during POST or PATCH. During GET, the returned value is rounded to the largest unit with a value greater than 1. |
| min_throughput_iops | integer | Specifies the minimum throughput in IOPS, 0 means none. Setting "min_throughput" is supported on AFF platforms only, unless FabricPool tiering policies are set. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when min_throughput is set during POST or PATCH. |
| min_throughput_mbps | integer | Specifies the minimum throughput in Megabytes per sec, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH.This cannot be set when min_throughput is set during POST or PATCH. |
| name | string | The QoS policy group name. This is mutually exclusive with UUID and other QoS attributes during POST and PATCH. |
| uuid | string | The QoS policy group UUID. This is mutually exclusive with name and other QoS attributes during POST and PATCH. |

iops_raw

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

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

latency_raw

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

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

throughput_raw

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

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

statistics

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

| Name | Type | Description |
|---|---|---|
| iops_raw | iops_raw | The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time. |
| latency_raw | latency_raw | The raw latency observed at the storage object, in microseconds. This can be divided by the raw IOPS value to calculate the average latency per I/O operation. |

| Name | Type | Description |
| --- | --- | --- |
| status | string | Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled with the next closest collection 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 does not have the latest data. |
| throughput_raw | throughput_raw | Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time. |
| timestamp | string | The timestamp of the performance data. |

svm

Required in POST

| Name | Type | Description |
| --- | --- | --- |
| _links | _links | |
| name | string | The name of the SVM. This field cannot be specified in a PATCH method. |
| uuid | string | The unique identifier of the SVM. This field cannot be specified in a PATCH method. |

user

The user set as owner of the qtree.

| Name | Type | Description |
|---|---|---|
| id | string | The numeric ID of the user who owns the qtree. Valid in POST or PATCH. |
| name | string | Alphanumeric username of user who owns the qtree. Valid in POST or PATCH. |

volume

Required in POST

| Name | Type | Description |
|---|---|---|
| _links | _links | |
| name | string | The name of the volume. This field cannot be specified in a PATCH method. |
| uuid | string | Unique identifier for the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move.<br>• example: 028baa66-41bd-11e9-81d5-00a0986138f7<br>• Introduced in: 9.6<br>• x-nullable: true |

qtree

A qtree is a directory at the top level of a volume to which a custom export policy (for fine-grained access control) and a quota rule can be applied, if required.

| Name | Type | Description |
|---|---|---|
| _links | _links | |
| _tags | array[string] | Tags are an optional way to track the uses of a resource. Tag values must be formatted as key:value strings. |

| Name | Type | Description |
| --- | --- | --- |
| export_policy | export_policy | Export Policy |
| ext_performance_monitoring | ext_performance_monitoring | |
| group | group | The user set as owner of the qtree. |
| id | integer | The identifier for the qtree, unique within the qtree's volume. |
| metric | metric | Performance numbers, such as IOPS latency and throughput. |
| name | string | The name of the qtree. Required in POST; optional in PATCH. |
| nas | nas | |
| path | string | Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH. This field is to be deprecated and replaced with nas.path. |
| qos_policy | qos_policy | When "min_throughput_iops", "min_throughput_mbps", "min_throughput", "max_throughput_iops", "max_throughput_mbps" or "max_throughput" attributes are specified, the storage object is assigned to an auto-generated QoS policy group. If the attributes are later modified, the auto-generated QoS policy-group attributes are modified. Attributes can be removed by specifying "0" and policy group by specifying "none". Upon deletion of the storage object or if the attributes are removed, then the QoS policy-group is also removed. |
| security_style | string | Security style. Valid in POST or PATCH. |

| Name | Type | Description |
|---|---|---|
| statistics | statistics | These are raw IOPS and throughput performance numbers. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster. |
| svm | svm | Required in POST |
| unix_permissions | integer | The UNIX permissions for the qtree. Valid in POST or PATCH. |
| user | user | The user set as owner of the qtree. |
| volume | volume | Required in POST |

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. |

# Create a qtree in a FlexVol or FlexGroup volume

POST `/storage/qtrees`

**Introduced In:** 9.6

Creates a qtree in a FlexVol volume or a FlexGroup volume.

After a qtree is created, the new qtree is assigned an identifier. This identifier is obtained using a qtree GET request. This identifier is used in the API path for the qtree PATCH and DELETE operations.

## Required properties

- `svm.uuid` or `svm.name` - Existing SVM in which to create the qtree.
- `volume.uuid` or `volume.name` - Existing volume in which to create the qtree.
- `name` - Name for the qtree.

## Recommended optional properties

If not specified in POST, the values are inherited from the volume.

- `security_style` - Security style for the qtree.
- `unix_permissions` - UNIX permissions for the qtree.
- `export_policy.name or export_policy.id` - Export policy of the SVM for the qtree.

## Related ONTAP commands

- `qtree create`

## Parameters

| Name | Type | In | Required | Description |
|---|---|---|---|---|
| return_timeout | integer | query | False | The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202.<br>• Default value: 0<br>• Max value: 120<br>• Min value: 0 |
| return_records | boolean | query | False | The default is false. If set to true, the records are returned.<br>• Default value: |

## Request Body

| Name | Type | Description |
|---|---|---|
| _tags | array[string] | Tags are an optional way to track the uses of a resource. Tag values must be formatted as key:value strings. |

| Name | Type | Description |
| --- | --- | --- |
| export_policy | export_policy | Export Policy |
| ext_performance_monitoring | ext_performance_monitoring | |
| group | group | The user set as owner of the qtree. |
| id | integer | The identifier for the qtree, unique within the qtree's volume. |
| name | string | The name of the qtree. Required in POST; optional in PATCH. |
| nas | nas | |
| path | string | Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH. This field is to be deprecated and replaced with nas.path. |
| qos_policy | qos_policy | When "min_throughput_iops", "min_throughput_mbps", "min_throughput", "max_throughput_iops", "max_throughput_mbps" or "max_throughput" attributes are specified, the storage object is assigned to an auto-generated QoS policy group. If the attributes are later modified, the auto-generated QoS policy-group attributes are modified. Attributes can be removed by specifying "0" and policy group by specifying "none". Upon deletion of the storage object or if the attributes are removed, then the QoS policy-group is also removed. |
| security_style | string | Security style. Valid in POST or PATCH. |
| svm | svm | Required in POST |
| unix_permissions | integer | The UNIX permissions for the qtree. Valid in POST or PATCH. |

| Name | Type | Description |
| --- | --- | --- |
| user | user | The user set as owner of the qtree. |
| volume | volume | Required in POST |

**Example request**


```
{
  "_tags": [
    "team:csi",
    "environment:test"
  ],
  "export_policy": {
    "id": 100,
    "name": "default"
  },
  "group": {
    "id": "20001",
    "name": "unix_group1"
  },
  "id": 1,
  "name": "string",
  "nas": {
    "path": "/volume3/qtree1"
  },
  "path": "/volume3/qtree1",
  "qos_policy": {
    "max_throughput": [
      "900KB/s",
      "500MB/s",
      "120GB/s",
      "5000IOPS",
      "5000IOPS,500KB/s",
      "2500IOPS,100MB/s",
      "1000IOPS,25MB/s"
    ],
    "max_throughput_iops": 10000,
    "max_throughput_mbps": 500,
    "min_throughput": [
      "900KB/s",
      "500MB/s",
      "120GB/s",
      "5000IOPS",
      "5000IOPS,500KB/s",
      "2500IOPS,100MB/s",
      "1000IOPS,25MB/s"
    ],
    "min_throughput_iops": 2000,
    "min_throughput_mbps": 500,
    "name": "performance",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
```

```
  },
  "security_style": "string",
  "svm": {
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  },
  "unix_permissions": 493,
  "user": {
    "id": "10001",
    "name": "unix_user1"
  },
  "volume": {
    "name": "volume1",
    "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
  }
}
```


## Response

```
Status: 202, Accepted
```

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

**Example response**


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


**Headers**

| Name | Description | Type |
|---|---|---|
| Location | Useful for tracking the resource location | string |

## Response

```
Status: 201, Created
```

## Error

```
Status: Default
```

ONTAP Error Response Codes

| Error Code | Description |
|---|---|
| 262245 | Invalid field value. |
| 262247 | Invalid field value. |
| 917525 | The specified volume does not exist in Vserver. |
| 917927 | The specified volume was not found. |
| 918232 | Either `volume.name` or `volume.uuid` must be provided. |
| 918236 | The specified `volume.uuid` and `volume.name` refer to different volumes. |
| 1703954 | Export Policy name specified is invalid. |
| 2621462 | The specified SVM does not exist. |
| 2621706 | The specified `svm.uuid` and `svm.name` do not refer to the same SVM. |
| 2621707 | No SVM was specified. Either `svm.name` or `svm.uuid` must be provided. |
| 5242881 | Cannot create qtree because the volume is read-only. |
| 5242886 | Failed to create qtree. |
| 5242894 | Qtree with empty name "" is not allowed, as that is reserved for the default qtree. |
| 5242900 | Qtree not supported on FlexCache volume |
| 5242948 | Qtree is not supported on FlexCache origin volume. |
| 5242951 | Export Policy supplied does not belong to the specified Export Policy ID. |
| 5242952 | Export Policy ID specified is invalid. |
| 5242953 | Qtree name must be provided. |
| 5242967 | UNIX user or group ID must be 32-bit unsigned integer. |
| 5242970 | FlexCache create is in progress for the volume. |

| Error Code | Description |
| --- | --- |
| 5242978 | The maximum number of qtrees for which extended performance monitoring can be enabled has been reached. Retry the POST request with ext_performance_monitoring.enabled set to false. |
| 5242979 | Qtree creation succeeded but failed to enable extended performance monitoring on the qtree. |
| 6622064 | Security-style NTFS is not supported on a SnapMirror active sync relationship volume. |
| 8454348 | QoS on qtrees is not supported because not all nodes in the cluster can support it. |
| 9437324 | The security style unified is not supported. |
| 23724050 | Failed to resolve user or group name. |
| 66846755 | Failed to determine whether volume is a FlexCache volume or not. |
| 66846839 | Failed to determine the effective cluster version of all the nodes hosting FlexCache volumes connected to FlexCache origin volume. |
| 92405926 | Qtree operation failed on the specified object store volume. |

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

## Definitions

**See Definitions**

href

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

_links

export_policy

Export Policy

| Name | Type | Description |
|---|---|---|
| id | integer | |
| name | string | |

ext_performance_monitoring

| Name | Type | Description |
|---|---|---|
| enabled | boolean | Specifies whether extended performance monitoring is enabled for the qtree. |

group

The user set as owner of the qtree.

| Name | Type | Description |
|---|---|---|
| id | string | The numeric ID of the group that owns the qtree. Valid in POST or PATCH. |
| name | string | Alphanumeric group name of group that owns the qtree. Valid in POST or PATCH. |

iops

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

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

latency

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

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

throughput

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

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

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

metric

Performance numbers, such as IOPS latency and throughput.

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

| 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 storage object. |
| timestamp | string | The timestamp of the performance data. |

nas

| Name | Type | Description |
|---|---|---|
| path | string | Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH. |

qos_policy

When "min_throughput_iops", "min_throughput_mbps", "min_throughput", "max_throughput_iops", "max_throughput_mbps" or "max_throughput" attributes are specified, the storage object is assigned to an auto-generated QoS policy group. If the attributes are later modified, the auto-generated QoS policy-group attributes are modified. Attributes can be removed by specifying "0" and policy group by specifying "none". Upon deletion of the storage object or if the attributes are removed, then the QoS policy-group is also removed.

| Name | Type | Description |
| --- | --- | --- |
| max_throughput | string | Specifies the maximum throughput in Kilobytes per sec, Megabytes per sec or Gigabytes per sec along with or without IOPS. 0 means none. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when either max_throughput_mbps or max_throughput_iops are set during POST or PATCH. During GET, the returned value is rounded to the largest unit with a value greater than 1. |
| max_throughput_iops | integer | Specifies the maximum throughput in IOPS, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when max_throughput is set during POST or PATCH. |
| max_throughput_mbps | integer | Specifies the maximum throughput in Megabytes per sec, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when max_throughput is set during POST or PATCH. |
| min_throughput | string | Specifies the minimum throughput in Kilobytes per sec, Megabytes per sec or Gigabytes per sec along with or without IOPS. 0 means none.Setting "min_throughput" is supported on AFF platforms only, unless FabricPool tiering policies are set. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when either min_throughput_mbps or min_throughput_iops are set during POST or PATCH. During GET, the returned value is rounded to the largest unit with a value greater than 1. |

| Name | Type | Description |
|---|---|---|
| min_throughput_iops | integer | Specifies the minimum throughput in IOPS, 0 means none. Setting "min_throughput" is supported on AFF platforms only, unless FabricPool tiering policies are set. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when min_throughput is set during POST or PATCH. |
| min_throughput_mbps | integer | Specifies the minimum throughput in Megabytes per sec, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH.This cannot be set when min_throughput is set during POST or PATCH. |
| name | string | The QoS policy group name. This is mutually exclusive with UUID and other QoS attributes during POST and PATCH. |
| uuid | string | The QoS policy group UUID. This is mutually exclusive with name and other QoS attributes during POST and PATCH. |

iops_raw

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

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

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

latency_raw

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

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

throughput_raw

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

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

statistics

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

| Name | Type | Description |
|---|---|---|
| iops_raw | iops_raw | The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time. |
| latency_raw | latency_raw | The raw latency observed at the storage object, in microseconds. This can be divided by the raw IOPS value to calculate the average latency per I/O operation. |
| status | string | Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled with the next closest collection 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 does not have the latest data. |
| throughput_raw | throughput_raw | Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time. |
| timestamp | string | The timestamp of the performance data. |

svm

Required in POST

| Name | Type | Description |
|---|---|---|
| name | string | The name of the SVM. This field cannot be specified in a PATCH method. |
| uuid | string | The unique identifier of the SVM. This field cannot be specified in a PATCH method. |

user

The user set as owner of the qtree.

| Name | Type | Description |
|---|---|---|
| id | string | The numeric ID of the user who owns the qtree. Valid in POST or PATCH. |
| name | string | Alphanumeric username of user who owns the qtree. Valid in POST or PATCH. |

volume

Required in POST

| Name | Type | Description |
|---|---|---|
| name | string | The name of the volume. This field cannot be specified in a PATCH method. |
| uuid | string | Unique identifier for the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move.<br>• example: 028baa66-41bd-11e9-81d5-00a0986138f7<br>• Introduced in: 9.6<br>• x-nullable: true |

qtree

A qtree is a directory at the top level of a volume to which a custom export policy (for fine-grained access control) and a quota rule can be applied, if required.

| Name | Type | Description |
|---|---|---|
| _tags | array[string] | Tags are an optional way to track the uses of a resource. Tag values must be formatted as key:value strings. |
| export_policy | export_policy | Export Policy |
| ext_performance_monitoring | ext_performance_monitoring | |
| group | group | The user set as owner of the qtree. |
| id | integer | The identifier for the qtree, unique within the qtree's volume. |
| name | string | The name of the qtree. Required in POST; optional in PATCH. |
| nas | nas | |
| path | string | Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH. This field is to be deprecated and replaced with nas.path. |
| qos_policy | qos_policy | When "min_throughput_iops", "min_throughput_mbps", "min_throughput", "max_throughput_iops", "max_throughput_mbps" or "max_throughput" attributes are specified, the storage object is assigned to an auto-generated QoS policy group. If the attributes are later modified, the auto-generated QoS policy-group attributes are modified. Attributes can be removed by specifying "0" and policy group by specifying "none". Upon deletion of the storage object or if the attributes are removed, then the QoS policy-group is also removed. |

| Name | Type | Description |
|---|---|---|
| security_style | string | Security style. Valid in POST or PATCH. |
| svm | svm | Required in POST |
| unix_permissions | integer | The UNIX permissions for the qtree. Valid in POST or PATCH. |
| user | user | The user set as owner of the qtree. |
| volume | volume | Required in POST |

job_link

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

error_arguments

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

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 qtree

DELETE `/storage/qtrees/{volume.uuid}/{id}`

**Introduced In:** 9.6

Deletes a qtree.

## Related ONTAP commands

- `qtree delete`

## Parameters

| Name | Type | In | Required | Description |
|---|---|---|---|---|
| volume.uuid | string | path | True | Volume UUID |
| id | string | path | True | Qtree ID |

| Name | Type | In | Required | Description |
|---|---|---|---|---|
| return_timeout | integer | query | False | The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202.<br>• Default value: 0<br>• Max value: 120<br>• Min value: 0 |

## Response

```
Status: 200, Ok
```

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

**Example response**


```
{
  "job": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "uuid": "string"
  }
}
```


## Response

```
Status: 202, Accepted
```

## Error

```
Status: Default
```

ONTAP Error Response Codes

| Error Code | Description |
|---|---|
| 917505 | Vserver not found. |
| 917506 | Volume not found. |
| 917525 | The specified volume does not exist in Vserver. |
| 918235 | A volume with UUID was not found. |
| 5242894 | The default qtree cannot be deleted. |
| 5242895 | Failed to delete the qtree. |
| 5242897 | This operation is not permitted on read-only volume. |
| 5242898 | This operation is only permitted on a data Vserver. |
| 5242916 | Cannot delete qtree because the volume contains one or more LUNs. |
| 5242925 | The limit for the number of concurrent delete jobs has been reached. |
| 5242927 | Unable to find qtree. |
| 5242955 | The UUID of the volume is required. |

| Error Code | Description |
|---|---|
| 5242957 | Failed to delete qtree with ID in the volume and SVM. |
| 5242965 | Invalid qtree path. The volume name component of the qtree path, must be the same as the volume specified with the parameter. |
| 10485796 | Cannot delete qtree because it contains a Storage Level Access Guard (SLAG). |

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 | |

job_link

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

error_arguments

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

returned_error

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

# Retrieve qtree properties

GET `/storage/qtrees/{volume.uuid}/{id}`

**Introduced In:** 9.6

Retrieves properties for a specific qtree identified by the `volume.uuid` and the `id` in the API path.

## Expensive properties

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

- `ext_performance_monitoring.enabled`
- `statistics.*`
- `metric.*`

## Related ONTAP commands

- `qtree show`

## Parameters

| Name | Type | In | Required | Description |
|---|---|---|---|---|
| volume.uuid | string | path | True | Volume UUID |
| id | string | path | True | Qtree ID |
| fields | array[string] | query | False | Specify the fields to return. |

## Response

```
Status: 200, Ok
```

| Name | Type | Description |
|---|---|---|
| _links | _links | |
| _tags | array[string] | Tags are an optional way to track the uses of a resource. Tag values must be formatted as key:value strings. |
| export_policy | export_policy | Export Policy |
| ext_performance_monitoring | ext_performance_monitoring | |
| group | group | The user set as owner of the qtree. |

| Name | Type | Description |
| --- | --- | --- |
| id | integer | The identifier for the qtree, unique within the qtree's volume. |
| metric | metric | Performance numbers, such as IOPS latency and throughput. |
| name | string | The name of the qtree. Required in POST; optional in PATCH. |
| nas | nas | |
| path | string | Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH. This field is to be deprecated and replaced with nas.path. |
| qos_policy | qos_policy | When "min_throughput_iops", "min_throughput_mbps", "min_throughput", "max_throughput_iops", "max_throughput_mbps" or "max_throughput" attributes are specified, the storage object is assigned to an auto-generated QoS policy group. If the attributes are later modified, the auto-generated QoS policy-group attributes are modified. Attributes can be removed by specifying "0" and policy group by specifying "none". Upon deletion of the storage object or if the attributes are removed, then the QoS policy-group is also removed. |
| security_style | string | Security style. Valid in POST or PATCH. |
| statistics | statistics | These are raw IOPS and throughput performance numbers. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster. |
| svm | svm | Required in POST |

| Name | Type | Description |
| --- | --- | --- |
| unix_permissions | integer | The UNIX permissions for the qtree. Valid in POST or PATCH. |
| user | user | The user set as owner of the qtree. |
| volume | volume | Required in POST |

**Example response**


```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "_tags": [
    "team:csi",
    "environment:test"
  ],
  "export_policy": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "id": 100,
    "name": "default"
  },
  "group": {
    "id": "20001",
    "name": "unix_group1"
  },
  "id": 1,
  "metric": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "duration": "PT5M",
    "iops": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "latency": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "status": "ok",
    "throughput": {
      "read": 200,
```

```
      "total": 1000,
      "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
  },
  "name": "string",
  "nas": {
    "path": "/volume3/qtree1"
  },
  "path": "/volume3/qtree1",
  "qos_policy": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "max_throughput": [
      "900KB/s",
      "500MB/s",
      "120GB/s",
      "5000IOPS",
      "5000IOPS,500KB/s",
      "2500IOPS,100MB/s",
      "1000IOPS,25MB/s"
    ],
    "max_throughput_iops": 10000,
    "max_throughput_mbps": 500,
    "min_throughput": [
      "900KB/s",
      "500MB/s",
      "120GB/s",
      "5000IOPS",
      "5000IOPS,500KB/s",
      "2500IOPS,100MB/s",
      "1000IOPS,25MB/s"
    ],
    "min_throughput_iops": 2000,
    "min_throughput_mbps": 500,
    "name": "performance",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "security_style": "string",
  "statistics": {
    "iops_raw": {
      "read": 200,
      "total": 1000,
```

```
      "write": 100
    },
    "latency_raw": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "status": "ok",
    "throughput_raw": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "timestamp": "2017-01-25 06:20:13 -0500"
  },
  "svm": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  },
  "unix_permissions": 493,
  "user": {
    "id": "10001",
    "name": "unix_user1"
  },
  "volume": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "volume1",
    "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
  }
}
```


## Error

```
Status: Default
```

ONTAP Error Response Codes

| Error Code | Description |
|---|---|
| 918235 | A volume with UUID was not found. |
| 2621462 | The specified SVM does not exist. |
| 5242889 | Failed to get the qtree from volume. |
| 5242956 | Failed to obtain a qtree with ID. |
| 5242965 | Invalid qtree path. The volume name component of the qtree path, must be the same as the volume specified with the parameter. |

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 | |

export_policy

Export Policy

| Name | Type | Description |
|---|---|---|
| _links | _links | |
| id | integer | |
| name | string | |

ext_performance_monitoring

| Name | Type | Description |
|---|---|---|
| enabled | boolean | Specifies whether extended performance monitoring is enabled for the qtree. |

group

The user set as owner of the qtree.

| Name | Type | Description |
|---|---|---|
| id | string | The numeric ID of the group that owns the qtree. Valid in POST or PATCH. |
| name | string | Alphanumeric group name of group that owns the qtree. Valid in POST or PATCH. |

iops

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

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

latency

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

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

throughput

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

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

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

metric

Performance numbers, such as IOPS latency and throughput.

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

| 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 storage object. |
| timestamp | string | The timestamp of the performance data. |

nas

| Name | Type | Description |
|---|---|---|
| path | string | Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH. |

qos_policy

When "min_throughput_iops", "min_throughput_mbps", "min_throughput", "max_throughput_iops", "max_throughput_mbps" or "max_throughput" attributes are specified, the storage object is assigned to an auto-generated QoS policy group. If the attributes are later modified, the auto-generated QoS policy-group attributes are modified. Attributes can be removed by specifying "0" and policy group by specifying "none". Upon deletion of the storage object or if the attributes are removed, then the QoS policy-group is also removed.

| Name | Type | Description |
| --- | --- | --- |
| _links | _links | |
| max_throughput | string | Specifies the maximum throughput in Kilobytes per sec, Megabytes per sec or Gigabytes per sec along with or without IOPS. 0 means none. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when either max_throughput_mbps or max_throughput_iops are set during POST or PATCH. During GET, the returned value is rounded to the largest unit with a value greater than 1. |
| max_throughput_iops | integer | Specifies the maximum throughput in IOPS, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when max_throughput is set during POST or PATCH. |
| max_throughput_mbps | integer | Specifies the maximum throughput in Megabytes per sec, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when max_throughput is set during POST or PATCH. |

| Name | Type | Description |
| --- | --- | --- |
| min_throughput | string | Specifies the minimum throughput in Kilobytes per sec, Megabytes per sec or Gigabytes per sec along with or without IOPS. 0 means none.Setting "min_throughput" is supported on AFF platforms only, unless FabricPool tiering policies are set. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when either min_throughput_mbps or min_throughput_iops are set during POST or PATCH. During GET, the returned value is rounded to the largest unit with a value greater than 1. |
| min_throughput_iops | integer | Specifies the minimum throughput in IOPS, 0 means none. Setting "min_throughput" is supported on AFF platforms only, unless FabricPool tiering policies are set. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when min_throughput is set during POST or PATCH. |
| min_throughput_mbps | integer | Specifies the minimum throughput in Megabytes per sec, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH.This cannot be set when min_throughput is set during POST or PATCH. |
| name | string | The QoS policy group name. This is mutually exclusive with UUID and other QoS attributes during POST and PATCH. |
| uuid | string | The QoS policy group UUID. This is mutually exclusive with name and other QoS attributes during POST and PATCH. |

iops_raw

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

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

latency_raw

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

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

throughput_raw

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

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

statistics

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

| Name | Type | Description |
|---|---|---|
| iops_raw | iops_raw | The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time. |
| latency_raw | latency_raw | The raw latency observed at the storage object, in microseconds. This can be divided by the raw IOPS value to calculate the average latency per I/O operation. |

| Name | Type | Description |
| --- | --- | --- |
| status | string | Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled with the next closest collection 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 does not have the latest data. |
| throughput_raw | throughput_raw | Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time. |
| timestamp | string | The timestamp of the performance data. |

svm

Required in POST

| Name | Type | Description |
| --- | --- | --- |
| _links | _links | |
| name | string | The name of the SVM. This field cannot be specified in a PATCH method. |
| uuid | string | The unique identifier of the SVM. This field cannot be specified in a PATCH method. |

user

The user set as owner of the qtree.

| Name | Type | Description |
|---|---|---|
| id | string | The numeric ID of the user who owns the qtree. Valid in POST or PATCH. |
| name | string | Alphanumeric username of user who owns the qtree. Valid in POST or PATCH. |

volume

Required in POST

| Name | Type | Description |
|---|---|---|
| _links | _links | |
| name | string | The name of the volume. This field cannot be specified in a PATCH method. |
| uuid | string | Unique identifier for the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move.<br>• example: 028baa66-41bd-11e9-81d5-00a0986138f7<br>• Introduced in: 9.6<br>• x-nullable: true |

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 properties for a qtree

PATCH `/storage/qtrees/{volume.uuid}/{id}`

**Introduced In:** 9.6

Updates properties for a specific qtree.

## Related ONTAP commands

- `qtree modify`
- `qtree rename`

## Parameters

| Name | Type | In | Required | Description |
|---|---|---|---|---|
| volume.uuid | string | path | True | Volume UUID |
| id | string | path | True | Qtree ID |

| Name | Type | In | Required | Description |
|---|---|---|---|---|
| return_timeout | integer | query | False | The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202.<br>• Default value: 0<br>• Max value: 120<br>• Min value: 0 |

## Request Body

| Name | Type | Description |
|---|---|---|
| _tags | array[string] | Tags are an optional way to track the uses of a resource. Tag values must be formatted as key:value strings. |
| export_policy | export_policy | Export Policy |
| ext_performance_monitoring | ext_performance_monitoring | |
| group | group | The user set as owner of the qtree. |

| Name | Type | Description |
| --- | --- | --- |
| id | integer | The identifier for the qtree, unique within the qtree's volume. |
| name | string | The name of the qtree. Required in POST; optional in PATCH. |
| nas | nas | |
| path | string | Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH. This field is to be deprecated and replaced with nas.path. |
| qos_policy | qos_policy | When "min_throughput_iops", "min_throughput_mbps", "min_throughput", "max_throughput_iops", "max_throughput_mbps" or "max_throughput" attributes are specified, the storage object is assigned to an auto-generated QoS policy group. If the attributes are later modified, the auto-generated QoS policy-group attributes are modified. Attributes can be removed by specifying "0" and policy group by specifying "none". Upon deletion of the storage object or if the attributes are removed, then the QoS policy-group is also removed. |
| security_style | string | Security style. Valid in POST or PATCH. |
| unix_permissions | integer | The UNIX permissions for the qtree. Valid in POST or PATCH. |
| user | user | The user set as owner of the qtree. |

**Example request**


```
{
  "_tags": [
    "team:csi",
    "environment:test"
  ],
  "export_policy": {
    "id": 100,
    "name": "default"
  },
  "group": {
    "id": "20001",
    "name": "unix_group1"
  },
  "id": 1,
  "name": "string",
  "nas": {
    "path": "/volume3/qtree1"
  },
  "path": "/volume3/qtree1",
  "qos_policy": {
    "max_throughput": [
      "900KB/s",
      "500MB/s",
      "120GB/s",
      "5000IOPS",
      "5000IOPS,500KB/s",
      "2500IOPS,100MB/s",
      "1000IOPS,25MB/s"
    ],
    "max_throughput_iops": 10000,
    "max_throughput_mbps": 500,
    "min_throughput": [
      "900KB/s",
      "500MB/s",
      "120GB/s",
      "5000IOPS",
      "5000IOPS,500KB/s",
      "2500IOPS,100MB/s",
      "1000IOPS,25MB/s"
    ],
    "min_throughput_iops": 2000,
    "min_throughput_mbps": 500,
    "name": "performance",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
```

```
  },
  "security_style": "string",
  "unix_permissions": 493,
  "user": {
    "id": "10001",
    "name": "unix_user1"
  }
}
```

## Response

```
Status: 200, Ok
```

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

**Example response**

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

## Response

```
Status: 202, Accepted
```

## Error

```
Status: Default
```

ONTAP Error Response Codes

| Error Code | Description |
|---|---|
| 262196 | The request contains a field which cannot be set in this operation. |
| 262278 | Required field is missing in the request. |

| Error Code | Description |
| --- | --- |
| 917505 | Vserver not found. |
| 917525 | The specified volume does not exist in Vserver. |
| 918235 | A volume with UUID was not found. |
| 5242887 | Failed to modify qtree. |
| 5242897 | This operation is not permitted on read-only volume. |
| 5242898 | This operation is only permitted on a data Vserver. |
| 5242902 | Missing inputs. |
| 5242915 | Failed to assign qtree export policy to qtree. |
| 5242927 | Unable to find qtree. |
| 5242945 | Failed to modify qtree. |
| 5242951 | Export policy supplied does not belong to the specified export policy ID. |
| 5242954 | Failed to get the qtree from volume. |
| 5242955 | The UUID of the volume is required. |
| 5242956 | Failed to obtain a qtree with ID. |
| 5242957 | Failed to delete the qtree. |
| 5242958 | Failed to rename the qtree with ID in the volume and SVM. |
| 5242959 | Successfully renamed qtree but the modify operation failed. |
| 5242965 | Invalid qtree path. The volume name component of the qtree path, must be the same as the volume specified with the parameter. |
| 5242967 | UNIX user or group ID must be 32-bit unsigned integer. |
| 5242971 | Qtree was renamed. However, the path modification failed. |
| 5242972 | Cannot rename qtree as that name already exists on a volume in the Vserver. |
| 5242973 | Cannot rename qtree to name with path concurrently on volume in Vserver, unless non-root qtrees in enabled on the volume. |
| 5242974 | Moved qtree. However, other properties were not modified. |
| 5242975 | Renamed qtree and moved the qtree. However, other properties were not modified. |
| 6622064 | Security-style NTFS is not supported on a SnapMirror active sync relationship volume. |

| Error Code | Description |
| --- | --- |
| 8454348 | QoS on qtrees is not supported because not all nodes in the cluster can support it. |
| 9437324 | The security style unified is not supported. |
| 23724050 | Failed to resolve user or group name. |

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

## Definitions

**See Definitions**

href

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

_links

export_policy

Export Policy

| Name | Type | Description |
|---|---|---|
| id | integer | |
| name | string | |

ext_performance_monitoring

| Name | Type | Description |
|---|---|---|
| enabled | boolean | Specifies whether extended performance monitoring is enabled for the qtree. |

group

The user set as owner of the qtree.

| Name | Type | Description |
|---|---|---|
| id | string | The numeric ID of the group that owns the qtree. Valid in POST or PATCH. |
| name | string | Alphanumeric group name of group that owns the qtree. Valid in POST or PATCH. |

iops

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

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

latency

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

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

throughput

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

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

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

metric

Performance numbers, such as IOPS latency and throughput.

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

| 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 storage object. |
| timestamp | string | The timestamp of the performance data. |

nas

| Name | Type | Description |
|---|---|---|
| path | string | Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH. |

qos_policy

When "min_throughput_iops", "min_throughput_mbps", "min_throughput", "max_throughput_iops", "max_throughput_mbps" or "max_throughput" attributes are specified, the storage object is assigned to an auto-generated QoS policy group. If the attributes are later modified, the auto-generated QoS policy-group attributes are modified. Attributes can be removed by specifying "0" and policy group by specifying "none". Upon deletion of the storage object or if the attributes are removed, then the QoS policy-group is also removed.

| Name | Type | Description |
| --- | --- | --- |
| max_throughput | string | Specifies the maximum throughput in Kilobytes per sec, Megabytes per sec or Gigabytes per sec along with or without IOPS. 0 means none. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when either max_throughput_mbps or max_throughput_iops are set during POST or PATCH. During GET, the returned value is rounded to the largest unit with a value greater than 1. |
| max_throughput_iops | integer | Specifies the maximum throughput in IOPS, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when max_throughput is set during POST or PATCH. |
| max_throughput_mbps | integer | Specifies the maximum throughput in Megabytes per sec, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when max_throughput is set during POST or PATCH. |
| min_throughput | string | Specifies the minimum throughput in Kilobytes per sec, Megabytes per sec or Gigabytes per sec along with or without IOPS. 0 means none.Setting "min_throughput" is supported on AFF platforms only, unless FabricPool tiering policies are set. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when either min_throughput_mbps or min_throughput_iops are set during POST or PATCH. During GET, the returned value is rounded to the largest unit with a value greater than 1. |

| Name | Type | Description |
|---|---|---|
| min_throughput_iops | integer | Specifies the minimum throughput in IOPS, 0 means none. Setting "min_throughput" is supported on AFF platforms only, unless FabricPool tiering policies are set. This is mutually exclusive with name and UUID during POST and PATCH. This cannot be set when min_throughput is set during POST or PATCH. |
| min_throughput_mbps | integer | Specifies the minimum throughput in Megabytes per sec, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH.This cannot be set when min_throughput is set during POST or PATCH. |
| name | string | The QoS policy group name. This is mutually exclusive with UUID and other QoS attributes during POST and PATCH. |
| uuid | string | The QoS policy group UUID. This is mutually exclusive with name and other QoS attributes during POST and PATCH. |

iops_raw

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

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

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

latency_raw

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

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

throughput_raw

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

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

statistics

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

| Name | Type | Description |
|---|---|---|
| iops_raw | iops_raw | The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time. |
| latency_raw | latency_raw | The raw latency observed at the storage object, in microseconds. This can be divided by the raw IOPS value to calculate the average latency per I/O operation. |
| status | string | Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled with the next closest collection 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 does not have the latest data. |
| throughput_raw | throughput_raw | Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time. |
| timestamp | string | The timestamp of the performance data. |

svm

Required in POST

| Name | Type | Description |
|---|---|---|
| name | string | The name of the SVM. This field cannot be specified in a PATCH method. |
| uuid | string | The unique identifier of the SVM. This field cannot be specified in a PATCH method. |

user

The user set as owner of the qtree.

| Name | Type | Description |
|---|---|---|
| id | string | The numeric ID of the user who owns the qtree. Valid in POST or PATCH. |
| name | string | Alphanumeric username of user who owns the qtree. Valid in POST or PATCH. |

volume

Required in POST

| Name | Type | Description |
|---|---|---|
| name | string | The name of the volume. This field cannot be specified in a PATCH method. |
| uuid | string | Unique identifier for the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move.<br>• example: 028baa66-41bd-11e9-81d5-00a0986138f7<br>• Introduced in: 9.6<br>• x-nullable: true |

qtree

A qtree is a directory at the top level of a volume to which a custom export policy (for fine-grained access control) and a quota rule can be applied, if required.

| Name | Type | Description |
|---|---|---|
| _tags | array[string] | Tags are an optional way to track the uses of a resource. Tag values must be formatted as key:value strings. |
| export_policy | export_policy | Export Policy |
| ext_performance_monitoring | ext_performance_monitoring | |
| group | group | The user set as owner of the qtree. |
| id | integer | The identifier for the qtree, unique within the qtree's volume. |
| name | string | The name of the qtree. Required in POST; optional in PATCH. |
| nas | nas | |
| path | string | Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH. This field is to be deprecated and replaced with nas.path. |
| qos_policy | qos_policy | When "min_throughput_iops", "min_throughput_mbps", "min_throughput", "max_throughput_iops", "max_throughput_mbps" or "max_throughput" attributes are specified, the storage object is assigned to an auto-generated QoS policy group. If the attributes are later modified, the auto-generated QoS policy-group attributes are modified. Attributes can be removed by specifying "0" and policy group by specifying "none". Upon deletion of the storage object or if the attributes are removed, then the QoS policy-group is also removed. |

| Name | Type | Description |
|---|---|---|
| security_style | string | Security style. Valid in POST or PATCH. |
| unix_permissions | integer | The UNIX permissions for the qtree. Valid in POST or PATCH. |
| user | user | The user set as owner of the qtree. |

job_link

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

error_arguments

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

returned_error

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

# Retrieve historical performance metrics for a qtree with extended performance monitoring enabled

GET `/storage/qtrees/{volume.uuid}/{qtree.id}/metrics`

**Introduced In:** 9.16

Retrieves historical performance metrics for a qtree which has extended performance monitoring enabled.

## Parameters

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

| Name | Type | In | Required | Description |
| --- | --- | --- | --- | --- |
| volume.name | string | query | False | Filter by volume.name |
| qtree.name | string | query | False | Filter by qtree.name |
| volume.uuid | string | path | True | Volume UUID |
| qtree.id | string | path | True | Qtree ID |
| interval | string | query | False | The time range for the data. Examples can be 1d, 1m, 1w, 1y. The period for each time range is as follows:<br>• 1d: Metrics over the most recent day sampled over 5 minutes.<br>• 1w: Metrics over the most recent week sampled over 30 minutes.<br>• 1m: Metrics over the most recent month sampled over 2 hours.<br>• 1y: Metrics over the most recent year sampled over a day.<br>• Default value: 1<br>• enum: ["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.<br>• Default value: 15<br>• Max value: 120<br>• 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": "PT5M",
      "iops": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "latency": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "qtree": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "id": 1,
        "name": "qt1"
      },
      "status": "ok",
      "svm": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
```

```
        "name": "svm1",
        "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
      },
      "throughput": {
        "read": 200,
        "total": 1000,
        "write": 100
      },
      "timestamp": "2017-01-25 06:20:13 -0500",
      "volume": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "volume1",
        "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
      }
    }
  ]
}
```


## Error

```
Status: Default
```

ONTAP Error Response Codes

| Error Code | Description |
|---|---|
| 918235 | A volume with UUID was not found. |
| 2621462 | The specified SVM does not exist. |
| 5242956 | Failed to obtain a qtree with ID. |
| 5242986 | Extended performance monitoring is not enabled on the qtree. |
| 8586227 | Interval not supported for this object. |

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 |
|---|---|---|
| next | href | |
| self | href | |

_links

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

iops

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

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

latency

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

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

qtree

| Name | Type | Description |
|---|---|---|
| _links | _links | |
| id | integer | The identifier for the qtree, unique within the qtree's volume. |
| name | string | The name of the qtree. |

svm

SVM, applies only to SVM-scoped objects.

| Name | Type | Description |
|---|---|---|
| _links | _links | |
| name | string | The name of the SVM. This field cannot be specified in a PATCH method. |
| uuid | string | The unique identifier of the SVM. This field cannot be specified in a PATCH method. |

throughput

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

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

volume

| Name | Type | Description |
|---|---|---|
| _links | _links | |
| name | string | The name of the volume. This field cannot be specified in a PATCH method. |
| uuid | string | Unique identifier for the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move.<br>• example: 028baa66-41bd-11e9-81d5-00a0986138f7<br>• Introduced in: 9.6<br>• x-nullable: true |

records

Performance numbers, such as IOPS latency and throughput.

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

| Name | Type | Description |
| --- | --- | --- |
| duration | string | The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations: |
| iops | iops | The rate of I/O operations observed at the storage object. |
| latency | latency | The round trip latency in microseconds observed at the storage object. |
| qtree | qtree | |
| 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. |
| svm | svm | SVM, applies only to SVM-scoped objects. |
| throughput | throughput | The rate of throughput bytes per second observed at the storage object. |

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

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. |

## Copyright information



## Trademark information