# Manage clusters

REST API reference

NetApp
September 12, 2025

# Table of Contents

# Manage clusters

## Cluster endpoint overview

## Overview

You can use this API to create a cluster, update cluster-wide configurations, and retrieve the current configuration details.

### Creating a cluster

You can create a new cluster by issuing a POST request to /cluster. Parameters are provided in the body of the POST request to configure cluster-wide settings and add nodes during the cluster setup.

#### Fields used for creating a cluster

The fields used for the cluster APIs fall into the following categories:

• Required cluster-wide configuration

• Optional cluster-wide configuration

#### Required cluster-wide configuration

The following fields are always required for any POST /cluster request:

• name

• password

#### Optional cluster-wide configuration

The following fields are used to set up additional cluster-wide configurations:

• location

• contact

• dns_domains

• name_servers

• ntp_servers

• timezone

• license

• configuration_backup

• management_interface

• nodes

#### Nodes field

The nodes field specifies the nodes to join to the cluster. To use this API, all nodes must run the same version of ONTAP. If you do not specify a node, the cluster is configured with one node added. The REST request is

issued to the node that is added to the cluster. If you specify one node, do not use the "node.cluster_interface.ip.address" field. If you specify multiple nodes, specify the node to which the REST request is issued in addition to the remote nodes. Use the "node.cluster_interface.ip.address" field to identify each node. All other node fields are optional in all cases. If you provide a field for one node, you need to provide the same field for all nodes.

### Node networking fields

The cluster management interface and each node management interface use the cluster management interface subnet mask and gateway. For advanced configurations in which the cluster and node management interfaces are on different subnets, use the /network/ip/interface APIs to configure network interfaces after setup is complete. The management interfaces are used to communicate with the name servers and NTP servers. The address family of the name servers and NTP servers must match the management interfaces address family.

### Single node cluster field

When the "single_node_cluster" field is set to "true", the cluster is created in single node cluster mode. You can provide a node field for this node for node-specific configuration but do not use the "node.cluster_interface.ip.address" field. Storage failover is configured to non-HA mode, and ports used for cluster ports are moved to the default IPspace. This might cause the node to reboot during setup. While a node reboots, the RESTful interface might not be available. See "Connection failures during cluster create" for more information.

### Create recommended aggregates parameter

When the "create_recommended_aggregates" parameter is set to "true", aggregates based on an optimal layout recommended by the system are created on each of the nodes in the cluster. The default setting is "false".

# Performance monitoring

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

# Monitoring cluster create status

### Errors before the job starts

Configuration in the POST /cluster request is validated before the cluster create job starts. If an invalid configuration is found, an HTTP error code in the 4xx range is returned. No cluster create job is started.

### Polling on the job

After a successful POST /cluster request is issued, an HTTP error code of 202 is returned along with a job UUID and link in the body of the response. The cluster create job continues asynchronously and is monitored with the job UUID using the /cluster/jobs API. The "message" field in the response of the GET /cluster/jobs/{uuid} request shows the current step in the job, and the "state" field shows the overall state of the

job.

**Errors during the job**

If a failure occurs during the cluster create job, the job body provides details of the error along with error code fields. See the error table under "Responses" in the POST /cluster documentation for common error codes and descriptions.

**Rerunning POST /cluster**

The POST /cluster request can be rerun if errors occur. When rerunning the request, use the same body and query parameters. You can change the value of any field in the original body or query, but you cannot change the provided fields. For example, an initial request might have a body section as follows:

```
body =
{
"name": "clusCreateRerun",
"password": "openSesame",
"nodes": [
  {
    "cluster_interface": {
      "ip": {
        "address": "1.1.1.1"
      }
    }
  },
  {
    "cluster_interface": {
      "ip": {
        "address": "2.2.2.2"
      }
    }
  }
]
}
```

A rerun request updates the body details to:

```
body =
{
"name": "clusCreateRerun",
"password": "openSesame",
"nodes": [
  {
    "cluster_interface": {
      "ip": {
        "address": "3.3.3.3"
      }
    }
  },
  {
    "cluster_interface": {
      "ip": {
        "address": "4.4.4.4"
      }
    }
  }
]
}
```

A rerun request with the following body details is invalid:

```
body =
{
"name": "clusCreateRerun",
"password": "openSesame",
"nodes": [
  {
    "cluster_interface": {
      "ip": {
        "address": "3.3.3.3"
      }
    }
  }
]
}
```

Note that the password might already be configured. If a password is already configured and then a new password is provided, the new request overwrites the existing password. If a password is already configured either by another interface or by a previous POST request to /cluster, authenticate any future REST requests with that password. If a POST request to /cluster with the default return_timeout of 0 returns an error, then the password was not changed.

**Connection failures during cluster create**

A request to poll the job status might fail during a cluster create job in the following two cases. In these cases, programmatic use of the RESTful interface might be resilient to these connection failures.

1. When the "single_node_cluster" flag is set to "true", the node might reboot. During this time, the RESTful interface might refuse connections and return errors on a GET request, or connection timeouts might occur. Programmatic use of the RESTful interface during reboots must consider these effects while polling a cluster create job.

2. The "mgmt_auto" LIF is removed during the cluster create job. A POST /cluster request might be issued on the "mgmt_auto" LIF. However, requests to poll the job status might fail during cluster create when the "mgmt_auto" LIF is removed. The "mgmt_auto" LIF is only removed if a cluster management interface is provided as an argument to POST /cluster, and only after the cluster management interface is created. Programmatic use of the POST /cluster API on the "mgmt_auto" LIF should be configured to dynamically switch to polling the job on the cluster management LIF.

## Modifying cluster configurations

The following fields can be used to modify a cluster-wide configuration:

- name
- location
- contact
- dns_domains
- name_servers
- timezone
- certificate

# Examples

**Minimally configuring a 2-node setup**

```
# Body
minimal_2_node_cluster.txt(body):
{
"name": "clusCreateExample1",
"password": "openSesame",
"nodes": [
  {
    "cluster_interface": {
      "ip": {
        "address": "1.1.1.1"
      }
    }
  },
  {
    "cluster_interface": {
      "ip": {
        "address": "2.2.2.2"
      }
    }
  }
]
}

# Request
curl -X POST "https://<mgmt-ip>/api/cluster" -d
"@minimal_2_node_cluster.txt"
```

**Setting up a single node with additional node configuration and auto aggregate creation**

```
# Body
single_node_additional_config.txt(body):
{
"name": "clusCreateExample2",
"password": "openSesame",
"nodes": [
  {
    "name": "singleNode",
    "location": "Sunnyvale"
  }
]
}

# Request
curl -X POST "https://<mgmt-
ip>/api/cluster?single_node_cluster=true&create_recommended_aggregates=tru
e" -d "@single_node_additional_config.txt"
```

**Modifying a cluster-wide configuration**

```
# Body
modify_cluster_config.txt(body):
{
"contact": "it@company.com",
"certificate": {
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
}

# Request
curl -X PATCH "https://<mgmt-ip>/api/cluster" -d
"@modify_cluster_config.txt"
```

# Creating a cluster using the cluster "create" operation

This example shows how to create a cluster using the cluster APIs. Specifically, this example shows the creation of a two-node cluster and uses information from the nodes themselves combined with user supplied information to configure the cluster.

**Preparing for setup**

Before the REST APIs can be issued to create the cluster, the cluster must be wired up and powered on. The network connections between the nodes for the cluster interconnect and the connections to the management network must be completed. After the nodes are powered on, the nodes automatically configure interfaces on the platform's default cluster ports to allow the nodes to discover each other during setup and expansion workflows. You must configure a management interface on one node or use the mgmt_auto LIF, which is assigned an IP address using DHCP, to start using the REST APIs. By making a console connection to a node, the cluster setup wizard guides you through the configuration of the initial node managment interface to which the REST calls can be sent. Once this step is completed, exit the wizard by typing "exit". You can then issue REST API requests.

1. Wire and power on the nodes.
2. Make a console connection to one node to access the cluster setup wizard.
3. Enter node management interface information to enable REST API requests to be sent to the node.

```
Welcome to the cluster setup wizard.
You can enter the following commands at any time:
"help" or "?" - if you want to have a question clarified,
"back" - if you want to change previously answered questions, and
"exit" or "quit" - if you want to quit the cluster setup wizard.
Any changes you made before quitting will be saved.
You can return to cluster setup at any time by typing "cluster setup".
To accept a default or omit a question, do not enter a value.
This system will send event messages and periodic reports to NetApp
Technical
Support. To disable this feature, enter
autosupport modify -support disable
within 24 hours.
Enabling AutoSupport can significantly speed problem determination and
resolution should a problem occur on your system.
For further information on AutoSupport, see:
  http://support.netapp.com/autosupport/
  Type yes to confirm and continue {yes}: yes
  Enter the node management interface port [e0c]:
    Enter the node management interface IP address: 10.224.82.249
    Enter the node management interface netmask: 255.255.192.0
    Enter the node management interface default gateway: 10.224.64.1
    A node management interface on port e0c with IP address 10.224.82.249
has been created.
    Use your web browser to complete cluster setup by accessing
    https://10.224.82.249
    Otherwise, press Enter to complete cluster setup using the command
line
    interface: exit
    Exiting the cluster setup wizard. Any changes you made have been
saved.
    The cluster administrator's account (username "admin") password is set
to the system default.
    Warning: You have exited the cluster setup wizard before completing
all
    of the tasks. The cluster is not configured. You can complete cluster
setup by typing
    "cluster setup" in the command line interface.
```

### Discovering the nodes

If you issue a GET /api/cluster/nodes request when the nodes are not in a cluster, the API returns a list of nodes that were discovered on the cluster interconnect. Information returned includes the node's serial number, model, software version, UUID, and cluster interface address. The number of nodes returned should

be the same as the number of nodes expected to be in the cluster. If too many nodes are discovered, remove the nodes that should not be part of the cluster. If not enough nodes are discovered, verify all the nodes are powered on, that the connections to the cluster interconnect are complete, and retry the command.

```
# The API:
/api/cluster/nodes

# The call:
curl -X GET "https://<mgmt-ip>/api/cluster/nodes?fields=state,uptime" -H
"accept: application/hal+json"

# The response:
{
"records": [
  {
    "uuid": "6dce4710-c860-11e9-b5bc-005056bb6135",
    "name": "cluster1",
    "uptime": 134555,
    "state": "up",
    "_links": {
      "self": {
        "href": "/api/cluster/nodes/6dce4710-c860-11e9-b5bc-005056bb6135"
      }
    }
  }
],
"num_records": 1,
"_links": {
  "self": {
    "href": "/api/cluster/nodes?fields=state,uptime"
  }
}
}
```

**Creating the cluster**

When the node information is available, including each node's cluster interface address, you can assemble the information for creating the cluster. Provide the cluster name and the password for the admin account. The rest of the information is optional and can be configured later using other APIs. Provide the cluster interface address for each node to be included in the cluster so that you can connect to it while adding it to the cluster. In addition to the cluster interface address, you can provide the optional node name, location, and management interface information. If you do not provide node names, nodes are named based on the cluster name. The nodes' managment interface subnet mask and gateway values are omitted and must be the same as the cluster management interface's subnet mask and gateway.

```
# The API:
/api/cluster

# The call:
curl -X POST "https://<mgmt-ip>/api/cluster" -H "accept:
application/hal+json" -H "accept: application/hal+json" -d
'{"name":"cluster1","location":"datacenter1","contact":"me","dns_domains":
["example.com"],"name_servers":["10.224.223.130","10.224.223.131","10.224.
223.132"],"ntp_servers":["time.nist.gov"],"management_interface":{"ip":{"a
ddress":"10.224.82.25","netmask":"255.255.192.0","gateway":"10.224.64.1"}}
,"password":"mypassword","license":{"keys":["AMEPOSOIKLKGEEEEDGNDEKSJDE"]}
,"nodes":[{"cluster_interface":{"ip":{"address":"169.254.245.113"}},"name"
:"node1","management_interface":{"ip":{"address":"10.224.82.29"}}},{"clust
er_interface":{"ip":{"address":"169.254.217.95"}},"name":"node2","manageme
nt_interface":{"ip":{"address":"10.224.82.31"}}}]}'

# The response:
{
"job": {
  "uuid": "b5bc07e2-19e9-11e9-a751-005056bbd95f",
  "_links": {
    "self": {
      "href": "/api/cluster/jobs/b5bc07e2-19e9-11e9-a751-005056bbd95f"
    }
  }
}
}
```

**Monitoring the progress of cluster creation**

To monitor the progress of the cluster create operation, poll the returned job link until the state value is no longer "running" or "queued".

```
# The API:
/api/cluster/jobs/b5bc07e2-19e9-11e9-a751-005056bbd95f

# The call:
curl -X GET "https://<mgmt-ip>/api/cluster/jobs/b5bc07e2-1e9-11e9-a751-
005056bbd95f" -H "accept: application/hal+json"

# The response:
{
"uuid": "b5bc07e2-19e9-11e9-a751-005056bbd95f",
"description": "POST /api/cluster",
"state": "success",
"message": "success",
"code": 0,
  "_links": {
    "self": {
      "href": "/api/cluster/jobs/b5bc07e2-19e9-11e9-a751-005056bbd95f"
  }
}
}
```

**Verifying the cluster information**

After the cluster is created, you can verify the information applied using a number of APIs. You can retrieve most of the information provided using the /api/cluster and /api/cluster/nodes APIs. In addition, you can view the network interface and route information using the /api/network APIs. The following example shows how to retrieve the cluster information:

```
# The API:
/api/cluster

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

# The response:
{
"name": "C1_sti44-vsim-ucs515w_1621957038",
"uuid": "5f7f57c7-bd67-11eb-95f4-005056a7b9b1",
"location": "sti",
"contact": "divyabha",
"version": {
  "full": "NetApp Release 9.10.1: Mon May 24 08:07:35 UTC 2021",
  "generation": 9,
```

```
      "major": 10,
      "minor": 1
    },
    "dns_domains": [
      "ctl.gdl.englab.netapp.com",
      "gdl.englab.netapp.com",
      "rtp.netapp.com",
      "eng.netapp.com",
      "netapp.com"
    ],
    "name_servers": [
      "10.224.223.131",
      "10.224.223.130"
    ],
    "ntp_servers": [
      "10.235.48.111"
    ],
    "peering_policy": {
      "minimum_passphrase_length": 8,
      "authentication_required": true,
      "encryption_required": false
    },
    "management_interfaces": [
      {
        "uuid": "beef2db7-bd67-11eb-95f4-005056a7b9b1",
        "name": "clus_mgmt",
        "ip": {
          "address": "10.236.153.229"
        },
        "_links": {
          "self": {
            "href": "/api/network/ip/interfaces/beef2db7-bd67-11eb-95f4-
005056a7b9b1"
          }
        }
      },
      {
        "uuid": "cb63e02c-bd72-11eb-95f4-005056a7b9b1",
        "name": "sti44-vsim-ucs515w_cluster_mgmt_inet6",
        "ip": {
          "address": "fd20:8b1e:b255:9051::a02"
        },
        "_links": {
          "self": {
            "href": "/api/network/ip/interfaces/cb63e02c-bd72-11eb-95f4-
005056a7b9b1"
```

```
        }
      }
    },
    {
      "uuid": "ea13dec1-bd72-11eb-bd00-005056a7f50e",
      "name": "sti44-vsim-ucs515x_cluster_mgmt_inet6",
      "ip": {
        "address": "fd20:8b1e:b255:9051::a0a"
      },
      "_links": {
        "self": {
          "href": "/api/network/ip/interfaces/ea13dec1-bd72-11eb-bd00-005056a7f50e"
        }
      }
    }
  ],
  "metric": {
    "timestamp": "2021-05-26T20:36:15Z",
    "duration": "PT15S",
    "status": "ok",
    "latency": {
      "other": 0,
      "total": 0,
      "read": 0,
      "write": 0
    },
    "iops": {
      "read": 0,
      "write": 0,
      "other": 0,
      "total": 0
    },
    "throughput": {
      "read": 0,
      "write": 0,
      "other": 0,
      "total": 0
    }
  },
  "statistics": {
    "timestamp": "2021-05-26T20:36:25Z",
    "status": "ok",
    "latency_raw": {
      "other": 0,
      "total": 0,
```

```
      "read": 0,
      "write": 0
    },
    "iops_raw": {
      "read": 0,
      "write": 0,
      "other": 0,
      "total": 0
    },
    "throughput_raw": {
      "read": 0,
      "write": 0,
      "other": 0,
      "total": 0
    }
  },
  "timezone": {
    "name": "America/New_York"
  },
  "certificate": {
    "uuid": "b282f3d1-bd67-11eb-95f4-005056a7b9b1",
    "_links": {
      "self": {
        "href": "/api/security/certificates/b282f3d1-bd67-11eb-95f4-
005056a7b9b1"
      }
    }
  },
  "san_optimized": false,
  "_links": {
    "self": {
      "href": "/api/cluster"
    }
  }
}
}

# The API:
/api/cluster

# The call:
curl -X GET "https://<mgmt-ip>/api/cluster?fields=management_interfaces"
-H "accept: application/hal+json"

# The response:
{
"management_interfaces": [
```

```
  {
    "uuid": "c661725a-19e9-11e9-a751-005056bbd95f",
    "name": "cluster_mgmt",
    "ip": {
      "address": "10.224.82.25"
    },
    "_links": {
      "self": {
        "href": "/api/network/ip/interfaces/c661725a-19e9-11e9-a751-
005056bbd95f"
      }
    }
  }
],
"_links": {
  "self": {
    "href": "/api/cluster"
  }
}
}
```

# Retrieve a cluster configuration

GET `/cluster`

**Introduced In:** 9.6

Retrieves the cluster configuration.

## Parameters

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

## Response

```
Status: 200, Ok
```

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

| Name | Type | Description |
|---|---|---|
| certificate | certificate | Support for this field will be removed in a future release. Please use /api/cluster/web for this field. Certificate used by cluster and node management interfaces for TLS connection requests. |
| contact | string | |
| dns_domains | array[string] | A list of DNS domains. Domain names have the following requirements:<br><br>• The name must contain only the following characters: A through Z, a through z, 0 through 9, ".", "-" or "_".<br><br>• The first character of each label, delimited by ".", must be one of the following characters: A through Z or a through z or 0 through 9.<br><br>• The last character of each label, delimited by ".", must be one of the following characters: A through Z, a through z, or 0 through 9.<br><br>• The top level domain must contain only the following characters: A through Z, a through z.<br><br>• The system reserves the following names:"all", "local", and "localhost". |
| location | string | |
| management_interfaces | array[management_interfaces] | |
| metric | metric | |
| name | string | |
| name_servers | array[string] | The list of IP addresses of the DNS servers. Addresses can be either IPv4 or IPv6 addresses. |
| ntp_servers | array[string] | Host name, IPv4 address, or IPv6 address for the external NTP time servers. |

| Name | Type | Description |
|---|---|---|
| peering_policy | peering_policy | |
| san_optimized | boolean | Specifies if this cluster is an All SAN Array. |
| statistics | statistics | |
| timezone | timezone | Provides the cluster-wide time zone information that localizes time found on messages displayed on each node's: <ul><li>console messages;</li><li>logging to internal ONTAP log files; and</li><li>localized REST API full ISO-8601 date, time, and time zone format information. Machine-to-machine interfaces, such as file access protocols (NFS, CIFS), block access protocols (SAN), and other protocols such as Manage ONTAP (ONTAPI), use second or subsecond time values that are based on world time or UTC.</li><li>Introduced in: 9.7</li></ul> |
| uuid | string | |
| version | version | This returns the cluster version information. When the cluster has more than one node, the cluster version is equivalent to the lowest of generation, major, and minor versions on all nodes. |

**Example response**

```json
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "certificate": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "cert1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "contact": "<a href="mailto:support@company.com">support@company.com</a>",
  "dns_domains": [
    "example.com",
    "example2.example3.com"
  ],
  "location": "building 1",
  "management_interfaces": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "ip": {
        "address": "10.10.10.7"
      },
      "name": "lif1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
  ],
  "metric": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "duration": "PT15S",
    "iops": {
```

```json
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "latency": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "status": "ok",
    "throughput": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "timestamp": "2017-01-25T11:20:13Z"
  },
  "name": "cluster1",
  "name_servers": [
    "10.224.65.20",
    "2001:db08:a0b:12f0::1"
  ],
  "ntp_servers": [
    "time.nist.gov",
    "10.98.19.20",
    "2610:20:6F15:15::27"
  ],
  "peering_policy": {
    "minimum_passphrase_length": 0
  },
  "statistics": {
    "iops_raw": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "latency_raw": {
      "read": 200,
      "total": 1000,
      "write": 100
    },
    "status": "ok",
    "throughput_raw": {
      "read": 200,
      "total": 1000,
      "write": 100
```

```
    },
    "timestamp": "2017-01-25T11:20:13Z"
  },
  "timezone": {
    "name": "America/New_York"
  },
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412",
  "version": {
    "full": "NetApp Release 9.4.0: Sun Nov 05 18:20:57 UTC 2017",
    "generation": 9,
    "major": 4,
    "minor": 0,
    "patch": "P2"
  }
}
```

## Error

```
Status: Default, Error
```

| Name | Type | Description |
|------|------|-------------|
| error | error | |

**Example error**

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

# Definitions

**See Definitions**

href

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

_links

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

certificate

Support for this field will be removed in a future release. Please use /api/cluster/web for this field.
Certificate used by cluster and node management interfaces for TLS connection requests.

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

configuration_backup

| Name | Type | Description |
|------|------|-------------|
| password | string | |
| url | string | An external backup location for the cluster configuration. This is mostly required for single node clusters where node and cluster configuration backups cannot be copied to other nodes in the cluster. |
| username | string | |
| validate_certificate | boolean | Use this parameter with the value "true" to validate the digital certificate of the remote server. Digital certificate validation is available only when the HTTPS protocol is used in the URL; it is disabled by default. |

license

License keys or NLF contents.

| Name | Type | Description |
|---|---|---|
| keys | array[string] | |

ip

Object to setup an interface along with its default router.

| Name | Type | Description |
|---|---|---|
| address | string | IPv4 or IPv6 address |
| gateway | string | The IPv4 or IPv6 address of the default router. |
| netmask | string | Input as netmask length (16) or IPv4 mask (255.255.0.0). For IPv6, the default value is 64 with a valid range of 1 to 127. Output is always netmask length. |

management_interface

The management interface of the cluster. The subnet mask and gateway for this interface are used for the node management interfaces provided in the node configuration.

| Name | Type | Description |
|---|---|---|
| ip | ip | Object to setup an interface along with its default router. |

ip

IP information

| Name | Type | Description |
|---|---|---|
| address | string | IPv4 or IPv6 address |

management_interfaces

| Name | Type | Description |
|---|---|---|
| _links | _links | |
| ip | ip | IP information |

| Name | Type | Description |
|------|------|-------------|
| name | string | The name of the interface. If only the name is provided, the SVM scope must be provided by the object this object is embedded in. |
| uuid | string | The UUID that uniquely identifies the interface. |

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 | Peformance 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. |

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

metric

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

| Name | Type | Description |
|------|------|-------------|
| status | string | 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. |

### node_setup_ip

The IP configuration for cluster setup.

| Name | Type | Description |
|------|------|-------------|
| address | string | IPv4 or IPv6 address |

### cluster_interface

The cluster network IP address of the node to be added.

### cluster_interfaces

Network interface

| Name | Type | Description |
| --- | --- | --- |
| _links | _links | |
| ip | ip | IP information |
| name | string | The name of the interface. If only the name is provided, the SVM scope must be provided by the object this object is embedded in. |
| uuid | string | The UUID that uniquely identifies the interface. |

cpu

CPU information.

| Name | Type | Description |
| --- | --- | --- |
| count | integer | Number of CPUs on the node. |
| firmware_release | string | Firmware release number. Defined by the CPU manufacturer. |
| processor | string | CPU type on the node. |

message

| Name | Type | Description |
| --- | --- | --- |
| code | string | Error code describing the current condition of chassis fans. |
| message | string | Message describing the current condition of chassis fans. It is only of use when `failed_fan.count` is not zero. |

failed_fan

| Name | Type | Description |
| --- | --- | --- |
| count | integer | Specifies a count of the number of chassis fans that are not operating within the recommended RPM range. |
| message | message | |

## message

| Name | Type | Description |
|---|---|---|
| code | string | Error code describing the current condition of power supply. |
| message | string | Message describing the state of any power supplies that are currently degraded. It is only of use when `failed_power_supply.count` is not zero. |

## failed_power_supply

| Name | Type | Description |
|---|---|---|
| count | integer | Number of failed power supply units. |
| message | [message](message) | |

## flash_cache

| Name | Type | Description |
|---|---|---|
| capacity | integer | Size in bytes |
| device_id | integer | |
| firmware_file | string | |
| firmware_version | string | |
| hardware_revision | string | |
| model | string | |
| part_number | string | |
| serial_number | string | |
| slot | string | |
| state | string | |

## frus

| Name | Type | Description |
|---|---|---|
| id | string | |
| state | string | |
| type | string | |

## controller

Controller information

| Name | Type | Description |
| --- | --- | --- |
| board | string | Type of the system board. This is defined by vendor. |
| cpu | cpu | CPU information. |
| failed_fan | failed_fan | |
| failed_power_supply | failed_power_supply | |
| flash_cache | array[flash_cache] | A list of Flash-Cache devices. Only returned when requested by name. |
| frus | array[frus] | List of FRUs on the node. Only returned when requested by name. |
| memory_size | integer | Memory available on the node, in bytes. |
| over_temperature | string | Specifies whether the hardware is currently operating outside of its recommended temperature range. The hardware shuts down if the temperature exceeds critical thresholds. |

## external_cache

Cache used for buffer management.

| Name | Type | Description |
| --- | --- | --- |
| is_enabled | boolean | Indicates whether the external cache is enabled. |
| is_hya_enabled | boolean | Indicates whether HyA caching is enabled. |
| is_rewarm_enabled | boolean | Indicates whether rewarm is enabled. |
| pcs_size | integer | PCS size in gigabytes. |

failure

Indicates the failure code and message.

| Name | Type | Description |
|------|------|-------------|
| code | integer | Message code |
| message | string | Detailed message based on the state. |

aggregate

Aggregate name and UUID.

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

error

Indicates the failed aggregate giveback code and message.

| Name | Type | Description |
|------|------|-------------|
| code | string | Message code. |
| message | string | Detailed message based on the state. |

status

| Name | Type | Description |
|------|------|-------------|
| aggregate | aggregate | Aggregate name and UUID. |
| error | error | Indicates the failed aggregate giveback code and message. |

| Name | Type | Description |
|------|------|-------------|
| state | string | Giveback state of the aggregate.<br><br>Possible values include no aggregates to giveback(nothing_to_giveback), failed to disable background disk firmware update(BDFU) on source node(failed_bdfu_source),<br><br>giveback delayed as disk firmware update is in progress on source node(delayed_bdfu_source), performing veto checks(running_checks). |

giveback

Represents the state of the node that is giving storage back to its HA partner.

| Name | Type | Description |
|------|------|-------------|
| failure | failure | Indicates the failure code and message. |
| state | string | |
| status | array[status] | Giveback status of each aggregate. |

interconnect

| Name | Type | Description |
|------|------|-------------|
| adapter | string | HA interconnect device name. |
| state | string | Indicates the HA interconnect status. |

partners

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

ports

| Name | Type | Description |
|------|------|-------------|
| number | integer | HA port number |
| state | string | HA port state:<br><br>• *down* - Logical HA link is down.<br><br>• *initialized* - Logical HA link is initialized. The physical link is up, but the subnet manager hasn't started to configure the port.<br><br>• *armed* - Logical HA link is armed. The physical link is up and the subnet manager started but did not yet complete configuring the port.<br><br>• *active* - Logical HA link is active.<br><br>• *reserved* - Logical HA link is active, but the physical link is down. |

takeover

This represents the state of the node that is taking over storage from its HA partner.

| Name | Type | Description |
|------|------|-------------|
| failure | failure | Indicates the failure code and message. |
| state | string | |

ha

| Name | Type | Description |
|------|------|-------------|
| auto_giveback | boolean | Specifies whether giveback is automatically initiated when the node that owns the storage is ready. |
| enabled | boolean | Specifies whether or not storage failover is enabled. |

| Name | Type | Description |
|------|------|-------------|
| giveback | giveback | Represents the state of the node that is giving storage back to its HA partner. |
| interconnect | interconnect | |
| partners | array[partners] | Nodes in this node's High Availability (HA) group. |
| ports | array[ports] | |
| takeover | takeover | This represents the state of the node that is taking over storage from its HA partner. |

local

| Name | Type | Description |
|------|------|-------------|
| ip | string | The hardware assist IP address. |
| port | integer | The hardware assist port. |
| state | string | The hardware assist monitor status. |

partner

| Name | Type | Description |
|------|------|-------------|
| ip | string | The hardware assist IP address. |
| port | integer | The hardware assist port. |
| state | string | The hardware assist monitor status. |

status

| Name | Type | Description |
|------|------|-------------|
| enabled | boolean | Indicates whether hardware assist is enabled on the node. |
| local | local | |
| partner | partner | |

hw_assist

The hardware assist information.

| Name | Type | Description |
|------|------|-------------|
| status | status | |

management_interface

The management interface of the node to be added. The subnet mask is set based on the management interface of the cluster or the managment interfaces of other nodes.

management_interfaces

Network interface

| Name | Type | Description |
|------|------|-------------|
| _links | _links | |
| ip | ip | IP information |
| name | string | The name of the interface. If only the name is provided, the SVM scope must be provided by the object this object is embedded in. |
| uuid | string | The UUID that uniquely identifies the interface. |

metric

CPU performance for the nodes.

| 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: |
| processor_utilization | integer | Average CPU Utilization for the node |

| 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. |
| timestamp | string | The timestamp of the performance data. |
| uuid | string | |

ports

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

metrocluster

Metrocluster

| Name | Type | Description |
|------|------|-------------|
| custom_vlan_capable | boolean | Indicates whether the MetroCluster over IP platform supports custom VLAN IDs. |
| ports | array[ports] | MetroCluster over IP ports. |

| Name | Type | Description |
|------|------|-------------|
| type | string | The Metrocluster configuration type |

nvram

| Name | Type | Description |
|------|------|-------------|
| battery_state | string | Specifies status of the NVRAM battery. Possible values:<br><br>• *battery_ok*<br>• *battery_partially_discharged*<br>• *battery_fully_discharged*<br>• *battery_not_present*<br>• *battery_near_end_of_life*<br>• *battery_at_end_of_life*<br>• *battery_unknown*<br>• *battery_over_charged*<br>• *battery_fully_charged* |
| id | integer | Vendor specific NVRAM ID of the node. |

api_service

Provides the properties of the service processor API service.

| Name | Type | Description |
|------|------|-------------|
| enabled | boolean | Indicates whether the service processor API service is enabled. |
| limit_access | boolean | Indicates whether the service processor API service limit access is enabled. |
| port | integer | Indicates the port number of service processor API service. |

auto_config

Provides the properties of the service processor auto configuration.

| Name | Type | Description |
|------|------|-------------|
| ipv4_subnet | string | Indicates the service processor auto configuration IPv4 subnet name. To enable IPv4 auto-config give the subnet name, give the value as null or an empty string "" to disable auto-config. |
| ipv6_subnet | string | Indicates the service processor auto configuration IPv6 subnet name. To enable IPv6 auto-config give the subnet name, give the value as null or an empty string "" to disable auto-config. |

backup

Provides the properties of the service processor backup partition.

| Name | Type | Description |
|------|------|-------------|
| is_current | boolean | Indicates whether the service processor is currently booted from the backup partition. |
| state | string | Status of the backup partition. |
| version | string | Firmware version of the backup partition. |

ipv4_interface

Object to setup an interface along with its default router.

| Name | Type | Description |
|------|------|-------------|
| address | string | IPv4 or IPv6 address |
| gateway | string | The IPv4 or IPv6 address of the default router. |
| netmask | string | Input as netmask length (16) or IPv4 mask (255.255.0.0). For IPv6, the default value is 64 with a valid range of 1 to 127. Output is always netmask length. |

ipv6_interface

Object to setup an interface along with its default router.

| Name | Type | Description |
|---|---|---|
| address | string | IPv6 address |
| gateway | string | The IPv6 address of the default router. |
| netmask | integer | The IPv6 netmask/prefix length. The default value is 64 with a valid range of 1 to 127. |

primary

Provides the properties of the service processor primary partition.

| Name | Type | Description |
|---|---|---|
| is_current | boolean | Indicates whether the service processor is currently booted from the primary partition. |
| state | string | Status of the primary partition. |
| version | string | Firmware version of the primary partition. |

ssh_info

Service processor SSH allowed IP address configuration applied across the cluster.

| Name | Type | Description |
|---|---|---|
| allowed_addresses | array[string] | Allowed IP addresses |

service_processor

| Name | Type | Description |
|---|---|---|
| api_service | api_service | Provides the properties of the service processor API service. |
| auto_config | auto_config | Provides the properties of the service processor auto configuration. |

| Name | Type | Description |
|---|---|---|
| autoupdate_enabled | boolean | Indicates whether the service processor can be automatically updated from ONTAP.<br><br>• Introduced in: 9.10<br><br>• x-ntap-readModify: true |
| backup | backup | Provides the properties of the service processor backup partition. |
| dhcp_enabled | boolean | Set to "true" to use DHCP to configure an IPv4 interface. Do not provide values for address, netmask and gateway when set to "true". |
| firmware_version | string | The version of firmware installed. |
| ipv4_interface | ipv4_interface | Object to setup an interface along with its default router. |
| ipv6_interface | ipv6_interface | Object to setup an interface along with its default router. |
| is_ip_configured | boolean | Indicates whether the service processor network is configured. |
| last_update_state | string | Provides the "update status" of the last service processor update. |
| link_status | string | |
| mac_address | string | |
| primary | primary | Provides the properties of the service processor primary partition. |
| ssh_info | ssh_info | Service processor SSH allowed IP address configuration applied across the cluster. |
| state | string | |
| type | string | |

statistics

Raw CPU performance for the nodes.

| Name | Type | Description |
|------|------|-------------|
| processor_utilization_base | integer | Base counter for CPU Utilization. |
| processor_utilization_raw | integer | Raw CPU Utilization for the node. This should be divided by the processor_utilization_base to calculate the percentage CPU utilization for the node. |
| 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. |
| timestamp | string | The timestamp of the performance data. |

version

This returns the cluster version information. When the cluster has more than one node, the cluster version is equivalent to the lowest of generation, major, and minor versions on all nodes.

| Name | Type | Description |
|------|------|-------------|
| full | string | The full cluster version string. |

| Name | Type | Description |
|---|---|---|
| generation | integer | The generation portion of the version. |
| major | integer | The major portion of the version. |
| minor | integer | The minor portion of the version. |
| patch | string | The patch portion of the version. |

vm

| Name | Type | Description |
|---|---|---|
| account_id | string | The cloud provider account ID. |
| deployment_id | string | The cloud provider deployment ID. |
| fault_domain | string | The VM fault domain. |
| instance_id | string | The cloud provider instance ID. |
| primary_ip | string | The VM primary IP address. |
| provider_type | string | Cloud provider where the VM is hosted. |
| update_domain | string | The VM update domain. |

nodes

Complete node information

| Name | Type | Description |
|---|---|---|
| _links | _links | |
| cluster_interfaces | array[cluster_interfaces] | |
| controller | controller | Controller information |

| Name | Type | Description |
|------|------|-------------|
| date | string | The current or "wall clock" time of the node in ISO-8601 date, time, and time zone format. The ISO-8601 date and time are localized based on the ONTAP cluster's timezone setting.<br><br>• example: 2019-04-17T11:49:26-04:00<br><br>• format: date-time<br><br>• readOnly: 1<br><br>• Introduced in: 9.6 |
| external_cache | external_cache | Cache used for buffer management. |
| ha | ha | |
| hw_assist | hw_assist | The hardware assist information. |
| is_all_flash_optimized | boolean | Specifies whether the node is all flash optimized. |
| is_all_flash_select_optimized | boolean | Specifies whether the node is all flash select optimized. |
| is_capacity_optimized | boolean | Specifies whether the node is capacity optimized. |
| is_performance_optimized | boolean | Specifies whether the node is performance optimized. |
| is_spares_low | boolean | Specifies whether or not the node is in spares low condition. |
| location | string | |
| management_interfaces | array[management_interfaces] | |

| Name | Type | Description |
|------|------|-------------|
| membership | string | Possible values:<br><br>• *available* - A node is detected on the internal cluster network and can be added to the cluster. Nodes that have a membership of "available" are not returned when a GET request is called when the cluster exists. Provide a query on the "membership" property for *available* to scan for nodes on the cluster network. Nodes that have a membership of "available" are returned automatically before a cluster is created.<br><br>• *joining* - Joining nodes are in the process of being added to the cluster. The node might be progressing through the steps to become a member or might have failed. The job to add the node or create the cluster provides details on the current progress of the node.<br><br>• *member* - Nodes that are members have successfully joined the cluster. |
| metric | metric | CPU performance for the nodes. |
| metrocluster | metrocluster | Metrocluster |
| model | string | |
| name | string | |
| nvram | nvram | |
| owner | string | Owner of the node. |
| serial_number | string | |
| service_processor | service_processor | |

| Name | Type | Description |
|------|------|-------------|
| state | string | State of the node:<br><br>• *up* - Node is up and operational.<br><br>• *booting* - Node is booting up.<br><br>• *down* - Node has stopped or is dumping core.<br><br>• *taken_over* - Node has been taken over by its HA partner and is not yet waiting for giveback.<br><br>• *waiting_for_giveback* - Node has been taken over by its HA partner and is waiting for the HA partner to giveback disks.<br><br>• *degraded* - Node has one or more critical services offline.<br><br>• *unknown* - Node or its HA partner cannot be contacted and there is no information on the node's state. |
| statistics | statistics | Raw CPU performance for the nodes. |
| storage_configuration | string | The storage configuration in the system. Possible values:<br><br>• *mixed_path*<br><br>• *single_path*<br><br>• *multi_path*<br><br>• *quad_path*<br><br>• *mixed_path_ha*<br><br>• *single_path_ha*<br><br>• *multi_path_ha*<br><br>• *quad_path_ha*<br><br>• *unknown* |
| system_id | string | |
| system_machine_type | string | OEM system machine type. |

| Name | Type | Description |
|---|---|---|
| uptime | integer | The total time, in seconds, that the node has been up. |
| uuid | string | |
| vendor_serial_number | string | OEM vendor serial number. |
| version | version | This returns the cluster version information. When the cluster has more than one node, the cluster version is equivalent to the lowest of generation, major, and minor versions on all nodes. |
| vm | vm | |

peering_policy

| Name | Type | Description |
|---|---|---|
| authentication_required | boolean | Indicates whether authentication is required in the communication between cluster peers. If true, authentication is required to establish communication between cluster peers. |
| encryption_required | boolean | Indicates whether encryption is required in the communication between cluster peers. If true, encryption is required to establish communication between cluster peers. |
| minimum_passphrase_length | integer | Minimum required length for a passphrase. For more information on password strength best practices, see: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html#implement-proper-password-strength-controls |

iops_raw

The number of I/O operations observed at the storage object. This can 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 | Peformance metric for write I/O operations. |

latency_raw

The raw latency in microseconds observed at the storage object. 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 | Peformance metric for write I/O operations. |

throughput_raw

Throughput bytes observed at the storage object. This can 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 | Peformance metric for write I/O operations. |

statistics

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

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

timezone

Provides the cluster-wide time zone information that localizes time found on messages displayed on each node's:

- console messages;

- logging to internal ONTAP log files; and

- localized REST API full ISO-8601 date, time, and time zone format information. Machine-to-machine interfaces, such as file access protocols (NFS, CIFS), block access protocols (SAN), and other protocols such as Manage ONTAP (ONTAPI), use second or subsecond time values that are based on world time or UTC.

| Name | Type | Description |
|---|---|---|
| name | string | The ONTAP time zone name or identification in either IANA time zone format "Area/Location", or an ONTAP traditional time zone.<br><br>The initial first node in cluster setting for time zone is "Etc/UTC". "Etc/UTC" is the IANA timezone "Area/Location" specifier for Coordinated Universal Time (UTC), which is an offset of 0.<br><br>**IANA time zone format**<br><br>The IANA time zone, formatted as "Area/Location", is based on geographic areas that have had the same time zone offset for many years.<br><br>"Location" represents a compound name using additional forward slashes.<br><br>An example of the "Area/Location" time zone is "America/New_York" and represents most of the United States Eastern Time Zone. Examples of "Area/Location" with "Location" as a compound name are "America/Argentina/Buenos_Aires" and "America/Indiana/Indianapolis".<br><br>**ONTAP traditional time zone**<br><br>Examples of the traditional time zones are "EST5EDT" for the United States Eastern Time Zone and "CET" for Central European Time Zone.<br><br>• example: America/New_York<br><br>• Introduced in: 9.7 |

error_arguments

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

error

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

# Update a cluster configuration

PATCH `/cluster`

**Introduced In:** 9.6

Updates the cluster configuration after the cluster is created.

## Related ONTAP commands

- `cluster identity modify`
- `system node modify`
- `vserver services dns modify`
- `vserver services name-service dns modify`
- `timezone`
- `security ssl modify`

## 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><br>• Default value: 1<br><br>• Max value: 120<br><br>• Min value: 0 |

## Request Body

| Name | Type | Description |
|------|------|-------------|
| _links | _links | |
| certificate | certificate | Support for this field will be removed in a future release. Please use /api/cluster/web for this field. Certificate used by cluster and node management interfaces for TLS connection requests. |
| contact | string | |

| Name | Type | Description |
|---|---|---|
| dns_domains | array[string] | A list of DNS domains. Domain names have the following requirements:<br><br>• The name must contain only the following characters: A through Z, a through z, 0 through 9, ".", "-" or "_".<br><br>• The first character of each label, delimited by ".", must be one of the following characters: A through Z or a through z or 0 through 9.<br><br>• The last character of each label, delimited by ".", must be one of the following characters: A through Z, a through z, or 0 through 9.<br><br>• The top level domain must contain only the following characters: A through Z, a through z.<br><br>• The system reserves the following names:"all", "local", and "localhost". |
| location | string | |
| management_interfaces | array[management_interfaces] | |
| name | string | |
| name_servers | array[string] | The list of IP addresses of the DNS servers. Addresses can be either IPv4 or IPv6 addresses. |
| peering_policy | peering_policy | |
| san_optimized | boolean | Specifies if this cluster is an All SAN Array. |

| Name | Type | Description |
|---|---|---|
| timezone | timezone | Provides the cluster-wide time zone information that localizes time found on messages displayed on each node's:<br><br>• console messages;<br><br>• logging to internal ONTAP log files; and<br><br>• localized REST API full ISO-8601 date, time, and time zone format information. Machine-to-machine interfaces, such as file access protocols (NFS, CIFS), block access protocols (SAN), and other protocols such as Manage ONTAP (ONTAPI), use second or subsecond time values that are based on world time or UTC.<br><br>• Introduced in: 9.7 |
| uuid | string | |

**Example request**

```json
{
  "certificate": {
    "name": "cert1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "contact": "<a href="mailto:support@company.com">support@company.com</a>",
  "dns_domains": [
    "example.com",
    "example2.example3.com"
  ],
  "location": "building 1",
  "management_interfaces": [
    {
      "name": "lif1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
  ],
  "name": "cluster1",
  "name_servers": [
    "10.224.65.20",
    "2001:db08:a0b:12f0::1"
  ],
  "peering_policy": {
    "minimum_passphrase_length": 0
  },
  "timezone": {
    "name": "America/New_York"
  },
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
}
```

## Response

```
Status: 202, Accepted
```

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

**Example response**

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

# Error

```
Status: Default
```

ONTAP Error Response Codes

| Error Code | Description |
| --- | --- |
| 3604491 | Updating timezone failed. |
| 3604520 | Internal error. System state is not correct to read or change timezone. |
| 8847361 | Too many DNS domains provided. |
| 8847362 | Too many name servers provided. |
| 9240587 | A name must be provided. |
| 12451843 | Certificate does not exist. |

# Definitions

**See Definitions**

href

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

_links

certificate

Support for this field will be removed in a future release. Please use /api/cluster/web for this field. Certificate used by cluster and node management interfaces for TLS connection requests.

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

configuration_backup

| Name | Type | Description |
|------|------|-------------|
| password | string | |
| url | string | An external backup location for the cluster configuration. This is mostly required for single node clusters where node and cluster configuration backups cannot be copied to other nodes in the cluster. |
| username | string | |
| validate_certificate | boolean | Use this parameter with the value "true" to validate the digital certificate of the remote server. Digital certificate validation is available only when the HTTPS protocol is used in the URL; it is disabled by default. |

license

License keys or NLF contents.

| Name | Type | Description |
|------|------|-------------|
| keys | array[string] | |

ip

Object to setup an interface along with its default router.

| Name | Type | Description |
|------|------|-------------|
| address | string | IPv4 or IPv6 address |
| gateway | string | The IPv4 or IPv6 address of the default router. |
| netmask | string | Input as netmask length (16) or IPv4 mask (255.255.0.0). For IPv6, the default value is 64 with a valid range of 1 to 127. Output is always netmask length. |

management_interface

The management interface of the cluster. The subnet mask and gateway for this interface are used for the node management interfaces provided in the node configuration.

| Name | Type | Description |
|------|------|-------------|
| ip | ip | Object to setup an interface along with its default router. |

ip

IP information

management_interfaces

| Name | Type | Description |
|------|------|-------------|
| _links | _links | |
| ip | ip | IP information |
| name | string | The name of the interface. If only the name is provided, the SVM scope must be provided by the object this object is embedded in. |
| uuid | string | The UUID that uniquely identifies the interface. |

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 | Peformance 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 | Peformance 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 | Peformance metric for write I/O operations. |

metric

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

| Name | Type | Description |
|---|---|---|
| status | string | 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. |

node_setup_ip

The IP configuration for cluster setup.

| Name | Type | Description |
|---|---|---|
| address | string | IPv4 or IPv6 address |

cluster_interface

The cluster network IP address of the node to be added.

cluster_interfaces

Network interface

| Name | Type | Description |
|---|---|---|
| ip | ip | IP information |
| name | string | The name of the interface. If only the name is provided, the SVM scope must be provided by the object this object is embedded in. |
| uuid | string | The UUID that uniquely identifies the interface. |

cpu

CPU information.

| Name | Type | Description |
|---|---|---|
| count | integer | Number of CPUs on the node. |
| firmware_release | string | Firmware release number. Defined by the CPU manufacturer. |
| processor | string | CPU type on the node. |

message

| Name | Type | Description |
|---|---|---|
| code | string | Error code describing the current condition of chassis fans. |
| message | string | Message describing the current condition of chassis fans. It is only of use when `failed_fan.count` is not zero. |

failed_fan

| Name | Type | Description |
|---|---|---|
| count | integer | Specifies a count of the number of chassis fans that are not operating within the recommended RPM range. |
| message | message | |

## message

| Name | Type | Description |
|------|------|-------------|
| code | string | Error code describing the current condition of power supply. |
| message | string | Message describing the state of any power supplies that are currently degraded. It is only of use when `failed_power_supply.count` is not zero. |

## failed_power_supply

| Name | Type | Description |
|------|------|-------------|
| count | integer | Number of failed power supply units. |
| message | message | |

## flash_cache

| Name | Type | Description |
|------|------|-------------|
| capacity | integer | Size in bytes |
| device_id | integer | |
| firmware_file | string | |
| firmware_version | string | |
| hardware_revision | string | |
| model | string | |
| part_number | string | |
| serial_number | string | |
| slot | string | |
| state | string | |

## frus

| Name | Type | Description |
|------|------|-------------|
| id | string | |
| state | string | |
| type | string | |

controller

Controller information

| Name | Type | Description |
|---|---|---|
| board | string | Type of the system board. This is defined by vendor. |
| cpu | cpu | CPU information. |
| failed_fan | failed_fan | |
| failed_power_supply | failed_power_supply | |
| flash_cache | array[flash_cache] | A list of Flash-Cache devices. Only returned when requested by name. |
| frus | array[frus] | List of FRUs on the node. Only returned when requested by name. |
| memory_size | integer | Memory available on the node, in bytes. |
| over_temperature | string | Specifies whether the hardware is currently operating outside of its recommended temperature range. The hardware shuts down if the temperature exceeds critical thresholds. |

external_cache

Cache used for buffer management.

failure

Indicates the failure code and message.

| Name | Type | Description |
|---|---|---|
| code | integer | Message code |
| message | string | Detailed message based on the state. |

aggregate

Aggregate name and UUID.

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

error

Indicates the failed aggregate giveback code and message.

| Name | Type | Description |
|------|------|-------------|
| code | string | Message code. |
| message | string | Detailed message based on the state. |

status

| Name | Type | Description |
|------|------|-------------|
| aggregate | aggregate | Aggregate name and UUID. |
| error | error | Indicates the failed aggregate giveback code and message. |
| state | string | Giveback state of the aggregate. Possible values include no aggregates to giveback(nothing_to_giveback), failed to disable background disk firmware update(BDFU) on source node(failed_bdfu_source), giveback delayed as disk firmware update is in progress on source node(delayed_bdfu_source), performing veto checks(running_checks). |

giveback

Represents the state of the node that is giving storage back to its HA partner.

| Name | Type | Description |
|------|------|-------------|
| failure | failure | Indicates the failure code and message. |
| state | string | |

| Name | Type | Description |
|------|------|-------------|
| status | array[status] | Giveback status of each aggregate. |

interconnect

| Name | Type | Description |
|------|------|-------------|
| adapter | string | HA interconnect device name. |
| state | string | Indicates the HA interconnect status. |

partners

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

ports

| Name | Type | Description |
|------|------|-------------|
| number | integer | HA port number |
| state | string | HA port state:<br><br>• *down* - Logical HA link is down.<br><br>• *initialized* - Logical HA link is initialized. The physical link is up, but the subnet manager hasn't started to configure the port.<br><br>• *armed* - Logical HA link is armed. The physical link is up and the subnet manager started but did not yet complete configuring the port.<br><br>• *active* - Logical HA link is active.<br><br>• *reserved* - Logical HA link is active, but the physical link is down. |

takeover

This represents the state of the node that is taking over storage from its HA partner.

| Name | Type | Description |
|------|------|-------------|
| failure | failure | Indicates the failure code and message. |
| state | string | |

ha

| Name | Type | Description |
|------|------|-------------|
| auto_giveback | boolean | Specifies whether giveback is automatically initiated when the node that owns the storage is ready. |
| enabled | boolean | Specifies whether or not storage failover is enabled. |
| giveback | giveback | Represents the state of the node that is giving storage back to its HA partner. |
| interconnect | interconnect | |
| partners | array[partners] | Nodes in this node's High Availability (HA) group. |
| ports | array[ports] | |
| takeover | takeover | This represents the state of the node that is taking over storage from its HA partner. |

local

| Name | Type | Description |
|------|------|-------------|
| ip | string | The hardware assist IP address. |
| port | integer | The hardware assist port. |
| state | string | The hardware assist monitor status. |

partner

| Name | Type | Description |
|------|------|-------------|
| ip | string | The hardware assist IP address. |
| port | integer | The hardware assist port. |
| state | string | The hardware assist monitor status. |

status

| Name | Type | Description |
|------|------|-------------|
| enabled | boolean | Indicates whether hardware assist is enabled on the node. |

hw_assist

The hardware assist information.

management_interface

The management interface of the node to be added. The subnet mask is set based on the management interface of the cluster or the managment interfaces of other nodes.

management_interfaces

Network interface

| Name | Type | Description |
|------|------|-------------|
| ip | ip | IP information |
| name | string | The name of the interface. If only the name is provided, the SVM scope must be provided by the object this object is embedded in. |
| uuid | string | The UUID that uniquely identifies the interface. |

metric

CPU performance for the nodes.

| 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: |
| processor_utilization | integer | Average CPU Utilization for the node |
| 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. |
| timestamp | string | The timestamp of the performance data. |
| uuid | string | |

ports

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

metrocluster

Metrocluster

| Name | Type | Description |
|------|------|-------------|
| custom_vlan_capable | boolean | Indicates whether the MetroCluster over IP platform supports custom VLAN IDs. |
| ports | array[ports] | MetroCluster over IP ports. |
| type | string | The Metrocluster configuration type |

nvram

| Name | Type | Description |
|------|------|-------------|
| battery_state | string | Specifies status of the NVRAM battery. Possible values: <br><br>• *battery_ok* <br>• *battery_partially_discharged* <br>• *battery_fully_discharged* <br>• *battery_not_present* <br>• *battery_near_end_of_life* <br>• *battery_at_end_of_life* <br>• *battery_unknown* <br>• *battery_over_charged* <br>• *battery_fully_charged* |
| id | integer | Vendor specific NVRAM ID of the node. |

api_service

Provides the properties of the service processor API service.

| Name | Type | Description |
|------|------|-------------|
| enabled | boolean | Indicates whether the service processor API service is enabled. |
| limit_access | boolean | Indicates whether the service processor API service limit access is enabled. |
| port | integer | Indicates the port number of service processor API service. |

### auto_config

Provides the properties of the service processor auto configuration.

| Name | Type | Description |
| --- | --- | --- |
| ipv4_subnet | string | Indicates the service processor auto configuration IPv4 subnet name. To enable IPv4 auto-config give the subnet name, give the value as null or an empty string "" to disable auto-config. |
| ipv6_subnet | string | Indicates the service processor auto configuration IPv6 subnet name. To enable IPv6 auto-config give the subnet name, give the value as null or an empty string "" to disable auto-config. |

### backup

Provides the properties of the service processor backup partition.

| Name | Type | Description |
| --- | --- | --- |
| is_current | boolean | Indicates whether the service processor is currently booted from the backup partition. |
| state | string | Status of the backup partition. |
| version | string | Firmware version of the backup partition. |

### ipv4_interface

Object to setup an interface along with its default router.

| Name | Type | Description |
| --- | --- | --- |
| address | string | IPv4 or IPv6 address |
| gateway | string | The IPv4 or IPv6 address of the default router. |

| Name | Type | Description |
|------|------|-------------|
| netmask | string | Input as netmask length (16) or IPv4 mask (255.255.0.0). For IPv6, the default value is 64 with a valid range of 1 to 127. Output is always netmask length. |

ipv6_interface

Object to setup an interface along with its default router.

| Name | Type | Description |
|------|------|-------------|
| address | string | IPv6 address |
| gateway | string | The IPv6 address of the default router. |
| netmask | integer | The IPv6 netmask/prefix length. The default value is 64 with a valid range of 1 to 127. |

primary

Provides the properties of the service processor primary partition.

| Name | Type | Description |
|------|------|-------------|
| is_current | boolean | Indicates whether the service processor is currently booted from the primary partition. |
| state | string | Status of the primary partition. |
| version | string | Firmware version of the primary partition. |

ssh_info

Service processor SSH allowed IP address configuration applied across the cluster.

| Name | Type | Description |
|------|------|-------------|
| allowed_addresses | array[string] | Allowed IP addresses |

service_processor

| Name | Type | Description |
|---|---|---|
| api_service | api_service | Provides the properties of the service processor API service. |
| auto_config | auto_config | Provides the properties of the service processor auto configuration. |
| autoupdate_enabled | boolean | Indicates whether the service processor can be automatically updated from ONTAP.<br><br>• Introduced in: 9.10<br><br>• x-ntap-readModify: true |
| backup | backup | Provides the properties of the service processor backup partition. |
| dhcp_enabled | boolean | Set to "true" to use DHCP to configure an IPv4 interface. Do not provide values for address, netmask and gateway when set to "true". |
| firmware_version | string | The version of firmware installed. |
| ipv4_interface | ipv4_interface | Object to setup an interface along with its default router. |
| ipv6_interface | ipv6_interface | Object to setup an interface along with its default router. |
| is_ip_configured | boolean | Indicates whether the service processor network is configured. |
| last_update_state | string | Provides the "update status" of the last service processor update. |
| link_status | string | |
| mac_address | string | |
| primary | primary | Provides the properties of the service processor primary partition. |

| Name | Type | Description |
|------|------|-------------|
| ssh_info | ssh_info | Service processor SSH allowed IP address configuration applied across the cluster. |
| state | string | |
| type | string | |

statistics

Raw CPU performance for the nodes.

| Name | Type | Description |
|------|------|-------------|
| processor_utilization_base | integer | Base counter for CPU Utilization. |
| processor_utilization_raw | integer | Raw CPU Utilization for the node. This should be divided by the processor_utilization_base to calculate the percentage CPU utilization for the node. |
| 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. |
| timestamp | string | The timestamp of the performance data. |

## version

This returns the cluster version information. When the cluster has more than one node, the cluster version is equivalent to the lowest of generation, major, and minor versions on all nodes.

| Name | Type | Description |
|------|------|-------------|
| full | string | The full cluster version string. |
| generation | integer | The generation portion of the version. |
| major | integer | The major portion of the version. |
| minor | integer | The minor portion of the version. |
| patch | string | The patch portion of the version. |

## vm

| Name | Type | Description |
|------|------|-------------|
| account_id | string | The cloud provider account ID. |
| deployment_id | string | The cloud provider deployment ID. |
| fault_domain | string | The VM fault domain. |
| instance_id | string | The cloud provider instance ID. |
| primary_ip | string | The VM primary IP address. |
| provider_type | string | Cloud provider where the VM is hosted. |
| update_domain | string | The VM update domain. |

## nodes

Complete node information

| Name | Type | Description |
|------|------|-------------|
| _links | _links | |
| cluster_interfaces | array[cluster_interfaces] | |
| controller | controller | Controller information |

| Name | Type | Description |
|------|------|-------------|
| date | string | The current or "wall clock" time of the node in ISO-8601 date, time, and time zone format. The ISO-8601 date and time are localized based on the ONTAP cluster's timezone setting.<br><br>• example: 2019-04-17T11:49:26-04:00<br>• format: date-time<br>• readOnly: 1<br>• Introduced in: 9.6 |
| ha | ha | |
| is_all_flash_optimized | boolean | Specifies whether the node is all flash optimized. |
| is_all_flash_select_optimized | boolean | Specifies whether the node is all flash select optimized. |
| is_capacity_optimized | boolean | Specifies whether the node is capacity optimized. |
| is_performance_optimized | boolean | Specifies whether the node is performance optimized. |
| is_spares_low | boolean | Specifies whether or not the node is in spares low condition. |
| location | string | |
| management_interfaces | array[management_interfaces] | |

| Name | Type | Description |
|---|---|---|
| membership | string | Possible values:<br><br>• *available* - A node is detected on the internal cluster network and can be added to the cluster. Nodes that have a membership of "available" are not returned when a GET request is called when the cluster exists. Provide a query on the "membership" property for *available* to scan for nodes on the cluster network. Nodes that have a membership of "available" are returned automatically before a cluster is created.<br><br>• *joining* - Joining nodes are in the process of being added to the cluster. The node might be progressing through the steps to become a member or might have failed. The job to add the node or create the cluster provides details on the current progress of the node.<br><br>• *member* - Nodes that are members have successfully joined the cluster. |
| metrocluster | metrocluster | Metrocluster |
| model | string | |
| name | string | |
| nvram | nvram | |
| owner | string | Owner of the node. |
| serial_number | string | |
| service_processor | service_processor | |

| Name | Type | Description |
|---|---|---|
| state | string | State of the node:<br><br>• *up* - Node is up and operational.<br><br>• *booting* - Node is booting up.<br><br>• *down* - Node has stopped or is dumping core.<br><br>• *taken_over* - Node has been taken over by its HA partner and is not yet waiting for giveback.<br><br>• *waiting_for_giveback* - Node has been taken over by its HA partner and is waiting for the HA partner to giveback disks.<br><br>• *degraded* - Node has one or more critical services offline.<br><br>• *unknown* - Node or its HA partner cannot be contacted and there is no information on the node's state. |
| storage_configuration | string | The storage configuration in the system. Possible values:<br><br>• *mixed_path*<br><br>• *single_path*<br><br>• *multi_path*<br><br>• *quad_path*<br><br>• *mixed_path_ha*<br><br>• *single_path_ha*<br><br>• *multi_path_ha*<br><br>• *quad_path_ha*<br><br>• *unknown* |
| system_id | string | |
| system_machine_type | string | OEM system machine type. |
| uptime | integer | The total time, in seconds, that the node has been up. |
| uuid | string | |

| Name | Type | Description |
|---|---|---|
| vendor_serial_number | string | OEM vendor serial number. |
| vm | vm | |

peering_policy

| Name | Type | Description |
|---|---|---|
| authentication_required | boolean | Indicates whether authentication is required in the communication between cluster peers. If true, authentication is required to establish communication between cluster peers. |
| encryption_required | boolean | Indicates whether encryption is required in the communication between cluster peers. If true, encryption is required to establish communication between cluster peers. |
| minimum_passphrase_length | integer | Minimum required length for a passphrase. For more information on password strength best practices, see: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html#implement-proper-password-strength-controls |

iops_raw

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

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

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

latency_raw

The raw latency in microseconds observed at the storage object. 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 | Peformance metric for write I/O operations. |

throughput_raw

Throughput bytes observed at the storage object. This can 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. |

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

statistics

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

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

timezone

Provides the cluster-wide time zone information that localizes time found on messages displayed on each node's:

- console messages;
- logging to internal ONTAP log files; and
- localized REST API full ISO-8601 date, time, and time zone format information. Machine-to-machine interfaces, such as file access protocols (NFS, CIFS), block access protocols (SAN), and other protocols such as Manage ONTAP (ONTAPI), use second or subsecond time values that are based on world time or UTC.

| Name | Type | Description |
|------|------|-------------|
| name | string | The ONTAP time zone name or identification in either IANA time zone format "Area/Location", or an ONTAP traditional time zone.<br><br>The initial first node in cluster setting for time zone is "Etc/UTC". "Etc/UTC" is the IANA timezone "Area/Location" specifier for Coordinated Universal Time (UTC), which is an offset of 0.<br><br>**IANA time zone format**<br><br>The IANA time zone, formatted as "Area/Location", is based on geographic areas that have had the same time zone offset for many years.<br><br>"Location" represents a compound name using additional forward slashes.<br><br>An example of the "Area/Location" time zone is "America/New_York" and represents most of the United States Eastern Time Zone. Examples of "Area/Location" with "Location" as a compound name are "America/Argentina/Buenos_Aires" and "America/Indiana/Indianapolis".<br><br>**ONTAP traditional time zone**<br><br>Examples of the traditional time zones are "EST5EDT" for the United States Eastern Time Zone and "CET" for Central European Time Zone.<br><br>• example: America/New_York<br><br>• Introduced in: 9.7 |

cluster

Complete cluster information

| Name | Type | Description |
|------|------|-------------|
| _links | _links | |
| certificate | certificate | Support for this field will be removed in a future release. Please use /api/cluster/web for this field. Certificate used by cluster and node management interfaces for TLS connection requests. |
| contact | string | |
| dns_domains | array[string] | A list of DNS domains. Domain names have the following requirements:<br><br>• The name must contain only the following characters: A through Z, a through z, 0 through 9, ".", "-" or "_".<br><br>• The first character of each label, delimited by ".", must be one of the following characters: A through Z or a through z or 0 through 9.<br><br>• The last character of each label, delimited by ".", must be one of the following characters: A through Z, a through z, or 0 through 9.<br><br>• The top level domain must contain only the following characters: A through Z, a through z.<br><br>• The system reserves the following names:"all", "local", and "localhost". |
| location | string | |
| management_interfaces | array[management_interfaces] | |
| name | string | |
| name_servers | array[string] | The list of IP addresses of the DNS servers. Addresses can be either IPv4 or IPv6 addresses. |
| peering_policy | peering_policy | |

| Name | Type | Description |
|------|------|-------------|
| san_optimized | boolean | Specifies if this cluster is an All SAN Array. |
| timezone | timezone | Provides the cluster-wide time zone information that localizes time found on messages displayed on each node's: <br><br> • console messages; <br><br> • logging to internal ONTAP log files; and <br><br> • localized REST API full ISO-8601 date, time, and time zone format information. Machine-to-machine interfaces, such as file access protocols (NFS, CIFS), block access protocols (SAN), and other protocols such as Manage ONTAP (ONTAPI), use second or subsecond time values that are based on world time or UTC. <br><br> • Introduced in: 9.7 |
| uuid | string | |

job_link

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

error_arguments

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

error

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

# Create a cluster

POST `/cluster`

**Introduced In:** 9.6

Creates a cluster.

## Required properties

- `name`
- `password`

## Recommended optional properties

- `location`
- `contact`
- `dns_domains`
- `name_servers`
- `ntp_servers`
- `license`
- `configuration_backup`
- `management_interface`
- `nodes`
- `timezone`

## Learn more

- DOC /cluster

## Parameters

| Name | Type | In | Required | Description |
|------|------|-----|----------|-------------|
| single_node_cluster | boolean | query | False | Configures a single node cluster. All cluster ports are reassigned to the default network. The storage failover settings are configured to non-HA. The node reboots during this operation. |
| create_recommended_aggregates | boolean | query | False | Create aggregates based on an optimal layout recommended by the system.<br><br>• Introduced in: 9.7<br><br>• Default value: |

| 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><br>• Default value: 1<br><br>• Max value: 120<br><br>• Min value: 0 |

## Request Body

| Name | Type | Description |
|------|------|-------------|
| _links | _links | |
| configuration_backup | configuration_backup | |
| contact | string | |

| Name | Type | Description |
|------|------|-------------|
| dns_domains | array[string] | A list of DNS domains. Domain names have the following requirements:<br><br>• The name must contain only the following characters: A through Z, a through z, 0 through 9, ".", "-" or "_".<br><br>• The first character of each label, delimited by ".", must be one of the following characters: A through Z or a through z or 0 through 9.<br><br>• The last character of each label, delimited by ".", must be one of the following characters: A through Z, a through z, or 0 through 9.<br><br>• The top level domain must contain only the following characters: A through Z, a through z.<br><br>• The system reserves the following names:"all", "local", and "localhost". |
| license | [license](#) | License keys or NLF contents. |
| location | string | |
| management_interface | [management_interface](#) | The management interface of the cluster. The subnet mask and gateway for this interface are used for the node management interfaces provided in the node configuration. |
| management_interfaces | array[[management_interfaces](#)] | |
| name | string | |
| name_servers | array[string] | The list of IP addresses of the DNS servers. Addresses can be either IPv4 or IPv6 addresses. |
| nodes | array[[nodes](#)] | |
| ntp_servers | array[string] | Host name, IPv4 address, or IPv6 address for the external NTP time servers. |

| Name | Type | Description |
|---|---|---|
| password | string | Initial admin password used to create the cluster. |
| peering_policy | peering_policy | |
| san_optimized | boolean | Specifies if this cluster is an All SAN Array. |
| timezone | timezone | Provides the cluster-wide time zone information that localizes time found on messages displayed on each node's:<br><br>• console messages;<br><br>• logging to internal ONTAP log files; and<br><br>• localized REST API full ISO-8601 date, time, and time zone format information. Machine-to-machine interfaces, such as file access protocols (NFS, CIFS), block access protocols (SAN), and other protocols such as Manage ONTAP (ONTAPI), use second or subsecond time values that are based on world time or UTC.<br><br>• Introduced in: 9.7 |
| uuid | string | |

**Example request**

```
{
  "configuration_backup": {
    "password": "yourpassword",
    "url": "http://10.224.65.198/backups",
    "username": "me"
  },
  "contact": "<a href="
mailto:support@company.com">support@company.com</a>",
  "dns_domains": [
    "example.com",
    "example2.example3.com"
  ],
  "license": {
    "keys": [
      "AMEPOSOIKLKGEEEEDGNDEKSJDE"
    ]
  },
  "location": "building 1",
  "management_interface": {
    "ip": {
      "address": "10.10.10.7",
      "gateway": "10.1.1.1",
      "netmask": "24"
    }
  },
  "management_interfaces": [
    {
      "name": "lif1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    }
  ],
  "name": "cluster1",
  "name_servers": [
    "10.224.65.20",
    "2001:db08:a0b:12f0::1"
  ],
  "nodes": [
    {
      "cluster_interface": {
        "ip": {
          "address": "10.10.10.7"
        }
      },
      "cluster_interfaces": [
```

```json
        {
          "name": "lif1",
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
        }
      ],
      "controller": {
        "board": "System Board XXVIII",
        "cpu": {
          "count": 20,
          "firmware_release": "string",
          "processor": "string"
        },
        "failed_fan": {
          "count": 1,
          "message": {
            "code": "111411207",
            "message": "There are no failed fans."
          }
        },
        "failed_power_supply": {
          "count": 1,
          "message": {
            "code": "111411208",
            "message": "There are no failed power supplies."
          }
        },
        "flash_cache": [
          {
            "capacity": 1024000000000,
            "device_id": 0,
            "firmware_file": "X9170_O000Z6300NVM",
            "firmware_version": "NA05",
            "hardware_revision": "A1",
            "model": "X1970A",
            "part_number": "119-00207",
            "serial_number": "A22P5061550000187",
            "slot": "6-1",
            "state": "string"
          }
        ],
        "frus": [
          {
            "id": "string",
            "state": "string",
            "type": "string"
          }
```

```
        ],
        "memory_size": 1024000000,
        "over_temperature": "string"
      },
      "date": "2019-04-17T11:49:26-04:00",
      "ha": {
        "giveback": {
          "failure": {
            "code": 852126,
            "message": "Failed to initiate giveback. Run the \"storage
failover show-giveback\" command for more information."
          },
          "state": "failed",
          "status": [
            {
              "aggregate": {
                "name": "aggr1",
                "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
              },
              "error": {
                "code": "852126",
                "message": "string"
              },
              "state": "string"
            }
          ]
        },
        "interconnect": {
          "adapter": "MVIA-RDMA",
          "state": "string"
        },
        "partners": [
          {
            "name": "node1",
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
          }
        ],
        "ports": [
          {
            "number": 0,
            "state": "active"
          }
        ],
        "takeover": {
          "failure": {
            "code": 852130,
```

```
            "message": "Failed to initiate takeover. Run the \"storage
failover show-takeover\" command for more information."
          },
          "state": "failed"
        }
      },
      "location": "rack 2 row 5",
      "management_interface": {
        "ip": {
          "address": "10.10.10.7"
        }
      },
      "management_interfaces": [
        {
          "name": "lif1",
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
        }
      ],
      "membership": "string",
      "metrocluster": {
        "ports": [
          {
            "name": "e1b"
          }
        ],
        "type": "string"
      },
      "model": "FAS3070",
      "name": "node-01",
      "nvram": {
        "battery_state": "string",
        "id": 0
      },
      "owner": "Example Corp",
      "serial_number": "4048820-60-9",
      "service_processor": {
        "api_service": {
          "port": 0
        },
        "auto_config": {
          "ipv4_subnet": "ipv4_mgmt",
          "ipv6_subnet": "ipv6_mgmt"
        },
        "autoupdate_enabled": null,
        "backup": {
          "state": "string",
```

```json
          "version": "11.6"
        },
        "dhcp_enabled": null,
        "firmware_version": "string",
        "ipv4_interface": {
          "address": "10.10.10.7",
          "gateway": "10.1.1.1",
          "netmask": "24"
        },
        "last_update_state": "string",
        "link_status": "string",
        "mac_address": "string",
        "primary": {
          "state": "string",
          "version": "11.6"
        },
        "ssh_info": {
          "allowed_addresses": [
            "10.10.10.7/24"
          ]
        },
        "state": "string",
        "type": "string"
      },
      "state": "string",
      "storage_configuration": "string",
      "system_id": "0537035403",
      "system_machine_type": "7Y56-CTOWW1",
      "uptime": 300536,
      "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412",
      "vendor_serial_number": "791603000068",
      "vm": {
        "account_id": "string",
        "deployment_id": "string",
        "fault_domain": "string",
        "instance_id": "string",
        "primary_ip": "string",
        "provider_type": "string",
        "update_domain": "string"
      }
    }
  ],
  "ntp_servers": [
    "time.nist.gov",
    "10.98.19.20",
    "2610:20:6F15:15::27"
```

```
    ],
    "password": "mypassword",
    "peering_policy": {
      "minimum_passphrase_length": 0
    },
    "timezone": {
      "name": "America/New_York"
    },
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  }
```

## Response

```
Status: 202, Accepted
```

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

**Example response**

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

## Error

```
Status: Default
```

ONTAP Error Response Codes

| Error Code | Description |
|------------|-------------|
| 262245 | The value provided is invalid. |
| 1179813 | Fields set for one node must be set for all nodes. |
| 1179817 | The IP address, subnet mask, and gateway must all be provided for cluster manangement interface. |
| 1179818 | The IP address and gateway must be of the same family. |

| Error Code | Description |
| --- | --- |
| 1179821 | An IP address and subnet mask conflicts with an existing entry. |
| 1179824 | An invalid gateway was provided. |
| 1179825 | All management and cluster config IP addresses must belong to the same address family. |
| 2097165 | An NTP server could not be reached. |
| 8847361 | Too many DNS domains provided. |
| 8847362 | Too many name servers provided. |
| 8847394 | An invalid DNS domain was provided. |
| 8978433 | An invalid license key was provided. |
| 9240587 | A name must be provided. |
| 9240594 | An invalid name was provided. |
| 39387137 | The URL provided is invalid. |
| 131727360 | A node could not be added to the cluster. This is a generic code, see response message for details. |
| 131727388 | Hostnames for NTP servers cannot be used without DNS configured. |
| 131727389 | URL and username are required for configuration backup. |

## Definitions

## See Definitions

### href

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

### _links

### certificate

Support for this field will be removed in a future release. Please use /api/cluster/web for this field. Certificate used by cluster and node management interfaces for TLS connection requests.

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

### configuration_backup

| Name | Type | Description |
|------|------|-------------|
| password | string | |
| url | string | An external backup location for the cluster configuration. This is mostly required for single node clusters where node and cluster configuration backups cannot be copied to other nodes in the cluster. |
| username | string | |
| validate_certificate | boolean | Use this parameter with the value "true" to validate the digital certificate of the remote server. Digital certificate validation is available only when the HTTPS protocol is used in the URL; it is disabled by default. |

### license

License keys or NLF contents.

| Name | Type | Description |
|------|------|-------------|
| keys | array[string] | |

ip

Object to setup an interface along with its default router.

| Name | Type | Description |
|------|------|-------------|
| address | string | IPv4 or IPv6 address |
| gateway | string | The IPv4 or IPv6 address of the default router. |
| netmask | string | Input as netmask length (16) or IPv4 mask (255.255.0.0). For IPv6, the default value is 64 with a valid range of 1 to 127. Output is always netmask length. |

management_interface

The management interface of the cluster. The subnet mask and gateway for this interface are used for the node management interfaces provided in the node configuration.

| Name | Type | Description |
|------|------|-------------|
| ip | ip | Object to setup an interface along with its default router. |

ip

IP information

management_interfaces

| Name | Type | Description |
|------|------|-------------|
| _links | _links | |
| ip | ip | IP information |
| name | string | The name of the interface. If only the name is provided, the SVM scope must be provided by the object this object is embedded in. |
| uuid | string | The UUID that uniquely identifies the interface. |

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 | Peformance 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 | Peformance 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 | Peformance metric for write I/O operations. |

metric

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

| Name | Type | Description |
|------|------|-------------|
| status | string | 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. |

node_setup_ip

The IP configuration for cluster setup.

| Name | Type | Description |
|------|------|-------------|
| address | string | IPv4 or IPv6 address |

cluster_interface

The cluster network IP address of the node to be added.

| Name | Type | Description |
|------|------|-------------|
| ip | node_setup_ip | The IP configuration for cluster setup. |

cluster_interfaces

Network interface

| Name | Type | Description |
|------|------|-------------|
| ip | ip | IP information |
| name | string | The name of the interface. If only the name is provided, the SVM scope must be provided by the object this object is embedded in. |
| uuid | string | The UUID that uniquely identifies the interface. |

cpu

CPU information.

| Name | Type | Description |
|------|------|-------------|
| count | integer | Number of CPUs on the node. |
| firmware_release | string | Firmware release number. Defined by the CPU manufacturer. |
| processor | string | CPU type on the node. |

message

| Name | Type | Description |
|------|------|-------------|
| code | string | Error code describing the current condition of chassis fans. |
| message | string | Message describing the current condition of chassis fans. It is only of use when `failed_fan.count` is not zero. |

failed_fan

| Name | Type | Description |
| --- | --- | --- |
| count | integer | Specifies a count of the number of chassis fans that are not operating within the recommended RPM range. |
| message | message | |

message

| Name | Type | Description |
| --- | --- | --- |
| code | string | Error code describing the current condition of power supply. |
| message | string | Message describing the state of any power supplies that are currently degraded. It is only of use when `failed_power_supply.count` is not zero. |

failed_power_supply

| Name | Type | Description |
| --- | --- | --- |
| count | integer | Number of failed power supply units. |
| message | message | |

flash_cache

| Name | Type | Description |
| --- | --- | --- |
| capacity | integer | Size in bytes |
| device_id | integer | |
| firmware_file | string | |
| firmware_version | string | |
| hardware_revision | string | |
| model | string | |
| part_number | string | |
| serial_number | string | |
| slot | string | |
| state | string | |

frus

| Name | Type | Description |
|------|------|-------------|
| id | string | |
| state | string | |
| type | string | |

controller

Controller information

| Name | Type | Description |
|------|------|-------------|
| board | string | Type of the system board. This is defined by vendor. |
| cpu | cpu | CPU information. |
| failed_fan | failed_fan | |
| failed_power_supply | failed_power_supply | |
| flash_cache | array[flash_cache] | A list of Flash-Cache devices. Only returned when requested by name. |
| frus | array[frus] | List of FRUs on the node. Only returned when requested by name. |
| memory_size | integer | Memory available on the node, in bytes. |
| over_temperature | string | Specifies whether the hardware is currently operating outside of its recommended temperature range. The hardware shuts down if the temperature exceeds critical thresholds. |

external_cache

Cache used for buffer management.

failure

Indicates the failure code and message.

| Name | Type | Description |
| --- | --- | --- |
| code | integer | Message code |
| message | string | Detailed message based on the state. |

aggregate

Aggregate name and UUID.

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

error

Indicates the failed aggregate giveback code and message.

| Name | Type | Description |
| --- | --- | --- |
| code | string | Message code. |
| message | string | Detailed message based on the state. |

status

| Name | Type | Description |
| --- | --- | --- |
| aggregate | aggregate | Aggregate name and UUID. |
| error | error | Indicates the failed aggregate giveback code and message. |

| Name | Type | Description |
|------|------|-------------|
| state | string | Giveback state of the aggregate.<br><br>Possible values include no aggregates to giveback(nothing_to_giveback), failed to disable background disk firmware update(BDFU) on source node(failed_bdfu_source),<br><br>giveback delayed as disk firmware update is in progress on source node(delayed_bdfu_source), performing veto checks(running_checks). |

giveback

Represents the state of the node that is giving storage back to its HA partner.

| Name | Type | Description |
|------|------|-------------|
| failure | failure | Indicates the failure code and message. |
| state | string | |
| status | array[status] | Giveback status of each aggregate. |

interconnect

| Name | Type | Description |
|------|------|-------------|
| adapter | string | HA interconnect device name. |
| state | string | Indicates the HA interconnect status. |

partners

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

ports

| Name | Type | Description |
|------|------|-------------|
| number | integer | HA port number |
| state | string | HA port state: <ul><li>*down* - Logical HA link is down.</li><li>*initialized* - Logical HA link is initialized. The physical link is up, but the subnet manager hasn't started to configure the port.</li><li>*armed* - Logical HA link is armed. The physical link is up and the subnet manager started but did not yet complete configuring the port.</li><li>*active* - Logical HA link is active.</li><li>*reserved* - Logical HA link is active, but the physical link is down.</li></ul> |

takeover

This represents the state of the node that is taking over storage from its HA partner.

| Name | Type | Description |
|------|------|-------------|
| failure | failure | Indicates the failure code and message. |
| state | string |  |

ha

| Name | Type | Description |
|------|------|-------------|
| auto_giveback | boolean | Specifies whether giveback is automatically initiated when the node that owns the storage is ready. |
| enabled | boolean | Specifies whether or not storage failover is enabled. |

| Name | Type | Description |
|---|---|---|
| giveback | giveback | Represents the state of the node that is giving storage back to its HA partner. |
| interconnect | interconnect | |
| partners | array[partners] | Nodes in this node's High Availability (HA) group. |
| ports | array[ports] | |
| takeover | takeover | This represents the state of the node that is taking over storage from its HA partner. |

local

| Name | Type | Description |
|---|---|---|
| ip | string | The hardware assist IP address. |
| port | integer | The hardware assist port. |
| state | string | The hardware assist monitor status. |

partner

| Name | Type | Description |
|---|---|---|
| ip | string | The hardware assist IP address. |
| port | integer | The hardware assist port. |
| state | string | The hardware assist monitor status. |

status

| Name | Type | Description |
|---|---|---|
| enabled | boolean | Indicates whether hardware assist is enabled on the node. |

hw_assist

The hardware assist information.

## management_interface

The management interface of the node to be added. The subnet mask is set based on the management interface of the cluster or the managment interfaces of other nodes.

| Name | Type | Description |
|------|------|-------------|
| ip | node_setup_ip | The IP configuration for cluster setup. |

management_interfaces

Network interface

| Name | Type | Description |
|------|------|-------------|
| ip | ip | IP information |
| name | string | The name of the interface. If only the name is provided, the SVM scope must be provided by the object this object is embedded in. |
| uuid | string | The UUID that uniquely identifies the interface. |

metric

CPU performance for the nodes.

| 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: |
| processor_utilization | integer | Average CPU Utilization for the node |

| 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. |
| timestamp | string | The timestamp of the performance data. |
| uuid | string | |

ports

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

metrocluster

Metrocluster

| Name | Type | Description |
|------|------|-------------|
| custom_vlan_capable | boolean | Indicates whether the MetroCluster over IP platform supports custom VLAN IDs. |
| ports | array[ports] | MetroCluster over IP ports. |

| Name | Type | Description |
|------|------|-------------|
| type | string | The Metrocluster configuration type |

nvram

| Name | Type | Description |
|------|------|-------------|
| battery_state | string | Specifies status of the NVRAM battery. Possible values:<br><br>• *battery_ok*<br>• *battery_partially_discharged*<br>• *battery_fully_discharged*<br>• *battery_not_present*<br>• *battery_near_end_of_life*<br>• *battery_at_end_of_life*<br>• *battery_unknown*<br>• *battery_over_charged*<br>• *battery_fully_charged* |
| id | integer | Vendor specific NVRAM ID of the node. |

api_service

Provides the properties of the service processor API service.

| Name | Type | Description |
|------|------|-------------|
| enabled | boolean | Indicates whether the service processor API service is enabled. |
| limit_access | boolean | Indicates whether the service processor API service limit access is enabled. |
| port | integer | Indicates the port number of service processor API service. |

auto_config

Provides the properties of the service processor auto configuration.

| Name | Type | Description |
|------|------|-------------|
| ipv4_subnet | string | Indicates the service processor auto configuration IPv4 subnet name. To enable IPv4 auto-config give the subnet name, give the value as null or an empty string "" to disable auto-config. |
| ipv6_subnet | string | Indicates the service processor auto configuration IPv6 subnet name. To enable IPv6 auto-config give the subnet name, give the value as null or an empty string "" to disable auto-config. |

backup

Provides the properties of the service processor backup partition.

| Name | Type | Description |
|------|------|-------------|
| is_current | boolean | Indicates whether the service processor is currently booted from the backup partition. |
| state | string | Status of the backup partition. |
| version | string | Firmware version of the backup partition. |

ipv4_interface

Object to setup an interface along with its default router.

| Name | Type | Description |
|------|------|-------------|
| address | string | IPv4 or IPv6 address |
| gateway | string | The IPv4 or IPv6 address of the default router. |
| netmask | string | Input as netmask length (16) or IPv4 mask (255.255.0.0). For IPv6, the default value is 64 with a valid range of 1 to 127. Output is always netmask length. |

ipv6_interface

Object to setup an interface along with its default router.

| Name | Type | Description |
|------|------|-------------|
| address | string | IPv6 address |
| gateway | string | The IPv6 address of the default router. |
| netmask | integer | The IPv6 netmask/prefix length. The default value is 64 with a valid range of 1 to 127. |

primary

Provides the properties of the service processor primary partition.

| Name | Type | Description |
|------|------|-------------|
| is_current | boolean | Indicates whether the service processor is currently booted from the primary partition. |
| state | string | Status of the primary partition. |
| version | string | Firmware version of the primary partition. |

ssh_info

Service processor SSH allowed IP address configuration applied across the cluster.

| Name | Type | Description |
|------|------|-------------|
| allowed_addresses | array[string] | Allowed IP addresses |

service_processor

| Name | Type | Description |
|------|------|-------------|
| api_service | api_service | Provides the properties of the service processor API service. |
| auto_config | auto_config | Provides the properties of the service processor auto configuration. |

| Name | Type | Description |
|---|---|---|
| backup | backup | Provides the properties of the service processor backup partition. |
| firmware_version | string | The version of firmware installed. |
| ipv4_interface | ipv4_interface | Object to setup an interface along with its default router. |
| is_ip_configured | boolean | Indicates whether the service processor network is configured. |
| last_update_state | string | Provides the "update status" of the last service processor update. |
| link_status | string | |
| mac_address | string | |
| primary | primary | Provides the properties of the service processor primary partition. |
| ssh_info | ssh_info | Service processor SSH allowed IP address configuration applied across the cluster. |
| state | string | |
| type | string | |

statistics

Raw CPU performance for the nodes.

| Name | Type | Description |
|---|---|---|
| processor_utilization_base | integer | Base counter for CPU Utilization. |
| processor_utilization_raw | integer | Raw CPU Utilization for the node. This should be divided by the processor_utilization_base to calculate the percentage CPU utilization for the node. |

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

## version

This returns the cluster version information. When the cluster has more than one node, the cluster version is equivalent to the lowest of generation, major, and minor versions on all nodes.

| Name | Type | Description |
|------|------|-------------|
| full | string | The full cluster version string. |
| generation | integer | The generation portion of the version. |
| major | integer | The major portion of the version. |
| minor | integer | The minor portion of the version. |
| patch | string | The patch portion of the version. |

## vm

| Name | Type | Description |
|------|------|-------------|
| account_id | string | The cloud provider account ID. |
| deployment_id | string | The cloud provider deployment ID. |
| fault_domain | string | The VM fault domain. |
| instance_id | string | The cloud provider instance ID. |
| primary_ip | string | The VM primary IP address. |
| provider_type | string | Cloud provider where the VM is hosted. |
| update_domain | string | The VM update domain. |

nodes

Complete node information

| Name | Type | Description |
|------|------|-------------|
| _links | _links | |
| cluster_interface | cluster_interface | The cluster network IP address of the node to be added. |
| cluster_interfaces | array[cluster_interfaces] | |
| controller | controller | Controller information |
| date | string | The current or "wall clock" time of the node in ISO-8601 date, time, and time zone format. The ISO-8601 date and time are localized based on the ONTAP cluster's timezone setting. <br><br> • example: 2019-04-17T11:49:26-04:00 <br><br> • format: date-time <br><br> • readOnly: 1 <br><br> • Introduced in: 9.6 |
| ha | ha | |

| Name | Type | Description |
|------|------|-------------|
| is_all_flash_optimized | boolean | Specifies whether the node is all flash optimized. |
| is_all_flash_select_optimized | boolean | Specifies whether the node is all flash select optimized. |
| is_capacity_optimized | boolean | Specifies whether the node is capacity optimized. |
| is_performance_optimized | boolean | Specifies whether the node is performance optimized. |
| is_spares_low | boolean | Specifies whether or not the node is in spares low condition. |
| location | string | |
| management_interface | management_interface | The management interface of the node to be added. The subnet mask is set based on the management interface of the cluster or the managment interfaces of other nodes. |
| management_interfaces | array[management_interfaces] | |

| Name | Type | Description |
|------|------|-------------|
| membership | string | Possible values:<br><br>• *available* - A node is detected on the internal cluster network and can be added to the cluster. Nodes that have a membership of "available" are not returned when a GET request is called when the cluster exists. Provide a query on the "membership" property for *available* to scan for nodes on the cluster network. Nodes that have a membership of "available" are returned automatically before a cluster is created.<br><br>• *joining* - Joining nodes are in the process of being added to the cluster. The node might be progressing through the steps to become a member or might have failed. The job to add the node or create the cluster provides details on the current progress of the node.<br><br>• *member* - Nodes that are members have successfully joined the cluster. |
| metrocluster | metrocluster | Metrocluster |
| model | string | |
| name | string | |
| nvram | nvram | |
| owner | string | Owner of the node. |
| serial_number | string | |
| service_processor | service_processor | |

| Name | Type | Description |
|---|---|---|
| state | string | State of the node: <br><br>• *up* - Node is up and operational. <br>• *booting* - Node is booting up. <br>• *down* - Node has stopped or is dumping core. <br>• *taken_over* - Node has been taken over by its HA partner and is not yet waiting for giveback. <br>• *waiting_for_giveback* - Node has been taken over by its HA partner and is waiting for the HA partner to giveback disks. <br>• *degraded* - Node has one or more critical services offline. <br>• *unknown* - Node or its HA partner cannot be contacted and there is no information on the node's state. |
| storage_configuration | string | The storage configuration in the system. Possible values: <br><br>• *mixed_path* <br>• *single_path* <br>• *multi_path* <br>• *quad_path* <br>• *mixed_path_ha* <br>• *single_path_ha* <br>• *multi_path_ha* <br>• *quad_path_ha* <br>• *unknown* |
| system_id | string | |
| system_machine_type | string | OEM system machine type. |
| uptime | integer | The total time, in seconds, that the node has been up. |
| uuid | string | |

| Name | Type | Description |
|------|------|-------------|
| vendor_serial_number | string | OEM vendor serial number. |
| vm | vm | |

peering_policy

| Name | Type | Description |
|------|------|-------------|
| authentication_required | boolean | Indicates whether authentication is required in the communication between cluster peers. If true, authentication is required to establish communication between cluster peers. |
| encryption_required | boolean | Indicates whether encryption is required in the communication between cluster peers. If true, encryption is required to establish communication between cluster peers. |
| minimum_passphrase_length | integer | Minimum required length for a passphrase. For more information on password strength best practices, see: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html#implement-proper-password-strength-controls |

iops_raw

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

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

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

latency_raw

The raw latency in microseconds observed at the storage object. 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 | Peformance metric for write I/O operations. |

throughput_raw

Throughput bytes observed at the storage object. This can 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. |

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

statistics

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

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

timezone

Provides the cluster-wide time zone information that localizes time found on messages displayed on each node's:

- console messages;
- logging to internal ONTAP log files; and
- localized REST API full ISO-8601 date, time, and time zone format information. Machine-to-machine interfaces, such as file access protocols (NFS, CIFS), block access protocols (SAN), and other protocols such as Manage ONTAP (ONTAPI), use second or subsecond time values that are based on world time or UTC.

| Name | Type | Description |
|------|------|-------------|
| name | string | The ONTAP time zone name or identification in either IANA time zone format "Area/Location", or an ONTAP traditional time zone.<br><br>The initial first node in cluster setting for time zone is "Etc/UTC". "Etc/UTC" is the IANA timezone "Area/Location" specifier for Coordinated Universal Time (UTC), which is an offset of 0.<br><br>**IANA time zone format**<br><br>The IANA time zone, formatted as "Area/Location", is based on geographic areas that have had the same time zone offset for many years.<br><br>"Location" represents a compound name using additional forward slashes.<br><br>An example of the "Area/Location" time zone is "America/New_York" and represents most of the United States Eastern Time Zone. Examples of "Area/Location" with "Location" as a compound name are "America/Argentina/Buenos_Aires" and "America/Indiana/Indianapolis".<br><br>**ONTAP traditional time zone**<br><br>Examples of the traditional time zones are "EST5EDT" for the United States Eastern Time Zone and "CET" for Central European Time Zone.<br><br>• example: America/New_York<br><br>• Introduced in: 9.7 |

cluster

Complete cluster information

| Name | Type | Description |
|------|------|-------------|
| _links | _links | |
| configuration_backup | configuration_backup | |
| contact | string | |
| dns_domains | array[string] | A list of DNS domains. Domain names have the following requirements:<br><br>• The name must contain only the following characters: A through Z, a through z, 0 through 9, ".", "-" or "_".<br><br>• The first character of each label, delimited by ".", must be one of the following characters: A through Z or a through z or 0 through 9.<br><br>• The last character of each label, delimited by ".", must be one of the following characters: A through Z, a through z, or 0 through 9.<br><br>• The top level domain must contain only the following characters: A through Z, a through z.<br><br>• The system reserves the following names:"all", "local", and "localhost". |
| license | license | License keys or NLF contents. |
| location | string | |
| management_interface | management_interface | The management interface of the cluster. The subnet mask and gateway for this interface are used for the node management interfaces provided in the node configuration. |
| management_interfaces | array[management_interfaces] | |
| name | string | |
| name_servers | array[string] | The list of IP addresses of the DNS servers. Addresses can be either IPv4 or IPv6 addresses. |

| Name | Type | Description |
|------|------|-------------|
| nodes | array[nodes] | |
| ntp_servers | array[string] | Host name, IPv4 address, or IPv6 address for the external NTP time servers. |
| password | string | Initial admin password used to create the cluster. |
| peering_policy | peering_policy | |
| san_optimized | boolean | Specifies if this cluster is an All SAN Array. |
| timezone | timezone | Provides the cluster-wide time zone information that localizes time found on messages displayed on each node's:<br><br>• console messages;<br><br>• logging to internal ONTAP log files; and<br><br>• localized REST API full ISO-8601 date, time, and time zone format information. Machine-to-machine interfaces, such as file access protocols (NFS, CIFS), block access protocols (SAN), and other protocols such as Manage ONTAP (ONTAPI), use second or subsecond time values that are based on world time or UTC.<br><br>• Introduced in: 9.7 |
| uuid | string | |

job_link

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

error_arguments

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

error

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

## Copyright information

## Trademark information