Manage cluster licensing

ONTAP 9.6 REST API reference

NetApp
August 29, 2024

Table of Contents

Manage cluster licensing

Cluster licensing licenses endpoint overview

Overview

Licensing allows you to tailor a system to meet an organization’s specific needs. New features can be enabled by purchasing a license from a NetApp sales associate. After installation of the license, the new feature is available immediately.

This interface manages licenses according to their supported feature. By default, the interface displays packages with installed licenses, but you can also return unlicensed packages.

Each feature has a compliance state which is indicated at the package level. Individual licenses also contain a compliance state indicated in the 'licenses' array. The state of the package is determined by analyzing the underlying licenses according to the following criteria:

- Licensing terms
- Cluster state

Licensing terms

The licensing terms define the conditions under which a package is considered 'compliant'. Individual licenses are evaluated based on the following:

- Scope
- Time period
- Usage

Scope

A package can be licensed under the following scopes:

- site
- cluster
- node

A package licensed under 'site' or 'cluster' permits the feature to be used by any node that is a member of the cluster.

A package licensed under 'node' scope permits the authorized node to use the feature. Within a cluster, if you haven’t supplied every node with a valid license, the package state will indicate 'noncompliant'. A license must be purchased for each node in a cluster for the package to be considered 'compliant'.

Time period

Some package licenses are only valid for a limited period of time. After a license has expired, the package state changes to 'noncompliant'. A new license will need to be purchased for the package to return to a 'compliant' state.

Usage

Some package licenses have additional terms that need to be maintained to keep a license in compliance. These conditions are defined by the individual license. For example, a license might define the maximum amount of storage that a node can allocate for the license to be 'compliant'.

Cluster state

A cluster's state consists of the following:

- Node online status
- Node cluster membership

Some features require that a node be online to display a valid compliance state. If a node can not be reached, or is not known to the cluster, the individual license may indicate an 'unknown' state.

Licensing keys

A license is issued in one of the following two formats:

- 26-character key
- NetApp License File (NLF)

The following is an example of a 26-character key:

```
  AMEPOSOIKLKGEEEEDGNDEKSJDE
```

The following is an example of a NLF key:

```
{
"statusResp": {
"version": "1",
"serialNumber": "123456789",
"message": "Success",
"licenses": {
  "capacity": "1",
  "type": "capacity",
  "licenseProtocol": "FABRICPOOL-TB",
  "package": "FabricPool",
  "licenseScope": "cluster"
},
"snStatus": "Active",
"product": "fabricpool",
"statusCode": "S007"
},
"Signature": "signatureABC"
}
```

Either format can be submitted, via this API, to enable features.

Examples

Retrieving a collection of licenses organized by package

This example retrieves a collection that contains one entry for each package (filtered to only the 'fabricpool' package).

```
# API
GET /cluster/licensing/licenses/?fields=*&name=fabricpool"

# Response
200 OK

# JSON Body
{
"records": [
{
  "name": "fabricpool",
  "scope": "cluster",
  "state": "compliant",
  "licenses": [
  {
    "owner": "testcluster-1",
    "serial_number": "4149027342",
    "state": "compliant",
    "capacity": {
    "maximum_size": 1099511627776,
    "used_size": 0
    }
  }
  ],
  "_links": {
  "self": {
    "href": "/api/cluster/licensing/licenses/fabricpool"
  }
  }
}
],
"num_records": 1,
"_links": {
"self": {
  "href": "/api/cluster/licensing/licenses/?fields=*&name=fabricpool"
}
}
}
```

Retrieving a collection of installed licenses

This example retrieves a collection containing all packages (except base) that have installed licenses.

```
# API
GET /cluster/licensing/licenses/?fields=*&name=!base
```

```
# Response
200 OK

# JSON Body
{
"records": [
{
  "name": "nfs",
  "scope": "node",
  "state": "compliant",
  "licenses": [
  {
    "owner": "testcluster-1",
    "serial_number": "1-81-0000000000000004149027492",
    "state": "compliant"
  }
  ],
  "_links": {
  "self": {
    "href": "/api/cluster/licensing/licenses/nfs"
  }
  }
},
{
  "name": "cifs",
  "scope": "node",
  "state": "compliant",
  "licenses": [
  {
    "owner": "testcluster-1",
    "serial_number": "1-81-0000000000000004149027492",
    "state": "compliant"
  }
  ],
  "_links": {
  "self": {
    "href": "/api/cluster/licensing/licenses/cifs"
  }
  }
}
],
"num_records": 2,
"_links": {
"self": {
  "href": "/api/cluster/licensing/licenses/?fields=*&name=!base"
```

```
}
}
}
```

Installing a NLF license

This example installs a single license in the NLF format.

> You must escape all double quotes and backslash characeters, of the JSON license, before it can be placed in the POST request.

```
# API
POST /cluster/licensing/licenses/

# JSON Body
{
"keys" : [ "{\"statusResp\":{\"snStatus\": \"Active\", \"licenses\":
{\"package\": \"FabricPool\", \"capacity\": \"1\", \"licenseProtocol\":
\"FABRICPOOL-TB\", \"type\": \"capacity\", \"licenseScope\": \"cluster\"},
\"message\": \"Success\", \"statusCode\": \"S007\", \"version\": \"1\",
\"product\": \"fabricpool\", \"serialNumber\": \"4149027342\"},
\"Signature\":\"SignatureABC\"}" ]
}

# Response
201 Created
```

Installing a 26-character key

This example installs a single 26-character key formatted license.

```
# API
POST /cluster/licensing/licenses/

# JSON Body
{
"keys" : [ "AAAAAAAAAAAAAAAAAAAAAAAAAAAA" ]
}

# Response
201 Created
```

Installing multiple licenses with one API call

This example demonstrates how multiple keys can be provided to install multiple features in a single API call.

```
# API
POST /cluster/licensing/licenses/

# JSON Body
{
"keys" : [ "AAAAAAAAAAAAAAAAAAAAAAAAAAAA",
           "BBBBBBBBBBBBBBBBBBBBBBBBBBBB" ]
}

# Response
201 Created
```

Retrieve license packages

GET `/cluster/licensing/licenses`

Retrieves a collection of license packages.

Related ONTAP commands

- `system license show-status`
- `system license show`

Learn more

- DOC /cluster/licensing/licenses

Parameters

Name	Type	In	Required	Description
scope	string	query	False	Filter by scope
licenses.expiry_time	string	query	False	Filter by licenses.expiry_time
licenses.compliance.state	string	query	False	Filter by licenses.compliance.state
licenses.capacity.used_size	integer	query	False	Filter by licenses.capacity.used_size

Name	Type	In	Required	Description
licenses.capacity.maximum_size	integer	query	False	Filter by licenses.capacity.maximum_size
licenses.serial_number	string	query	False	Filter by licenses.serial_number
licenses.active	boolean	query	False	Filter by licenses.active
licenses.owner	string	query	False	Filter by licenses.owner
licenses.evaluation	boolean	query	False	Filter by licenses.evaluation
licenses.start_time	string	query	False	Filter by licenses.start_time
state	string	query	False	Filter by state
name	string	query	False	Filter by name
fields	array[string]	query	False	Specify the fields to return.
max_records	integer	query	False	Limit the number of records returned.
return_records	boolean	query	False	The default is true for GET calls. When set to false, only the number of records is returned.

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

Response

```
Status: 200, Ok
```

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

Example response


```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "keys": [
        "AMEPOSOIKLKGEEEEDGNDEKSJDE"
      ],
      "licenses": [
        {
          "capacity": {
            "maximum_size": 0,
            "used_size": 0
          },
          "compliance": {
            "state": "compliant"
          },
          "expiry_time": "2019-03-02 19:00:00 UTC",
          "owner": "cluster1",
          "serial_number": "123456789",
          "start_time": "2019-02-02 19:00:00 UTC"
        }
      ],
      "name": "NFS",
      "scope": "string",
      "state": "compliant"
    }
  ]
}
```

Error

```
Status: Default, Error
```

Name	Type	Description
error	error	

Example error


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


Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

_links

Name	Type	Description
self	href	

capacity

Name	Type	Description
maximum_size	integer	Licensed capacity size (in bytes) that can be used.
used_size	integer	Capacity that is currently used (in bytes).

compliance

Name	Type	Description
state	string	Compliance state of the license.

licenses

Name	Type	Description
active	boolean	A flag indicating whether the license is currently being enforced.
capacity	capacity	
compliance	compliance	
evaluation	boolean	A flag indicating whether the license is in evaluation mode.

Name	Type	Description
expiry_time	string	Date and time when the license expires.
owner	string	Cluster, node or license manager that owns the license.
serial_number	string	Serial number of the license.
start_time	string	Date and time when the license starts.

records

Name	Type	Description
_links	_links	
keys	array[string]	
licenses	array[licenses]	Installed licenses of the package.
name	string	Name of the license.
scope	string	Scope of the license.
state	string	Summary state of package based on all installed licenses.

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

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

Install one or more feature licenses

POST `/cluster/licensing/licenses`

Installs one or more feature licenses.

Required properties

- `keys` - Array containing a list of NLF or 26-character license keys.

Related ONTAP commands

- `system license add`

Learn more

- DOC /cluster/licensing/licenses

Parameters

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

Request Body

Name	Type	Description
_links	_links	
keys	array[string]	
licenses	array[licenses]	Installed licenses of the package.
name	string	Name of the license.
scope	string	Scope of the license.
state	string	Summary state of package based on all installed licenses.

Example request


```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "keys": [
    "AMEPOSOIKLKGEEEEDGNDEKSJDE"
  ],
  "licenses": [
    {
      "capacity": {
        "maximum_size": 0,
        "used_size": 0
      },
      "compliance": {
        "state": "compliant"
      },
      "expiry_time": "2019-03-02 19:00:00 UTC",
      "owner": "cluster1",
      "serial_number": "123456789",
      "start_time": "2019-02-02 19:00:00 UTC"
    }
  ],
  "name": "NFS",
  "scope": "string",
  "state": "compliant"
}
```


Response

```
Status: 201, Created
```

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

Example response


```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "keys": [
        "AMEPOSOIKLKGEEEEDGNDEKSJDE"
      ],
      "licenses": [
        {
          "capacity": {
            "maximum_size": 0,
            "used_size": 0
          },
          "compliance": {
            "state": "compliant"
          },
          "expiry_time": "2019-03-02 19:00:00 UTC",
          "owner": "cluster1",
          "serial_number": "123456789",
          "start_time": "2019-02-02 19:00:00 UTC"
        }
      ],
      "name": "NFS",
      "scope": "string",
      "state": "compliant"
    }
  ]
}
```

Error

```
Status: Default
```

ONTAP Error Response Codes

Error Code	Description
1115117	Generic licensing error
1115122	No cluster serial number found
1115124	No node serial number found
1115130	No license code was provided
1115131	Installation of the license failed
1115132	License already exists on system
1115134	Serial number does not belong to node
1115141	License data is invalid
1115142	License signature is invalid
1115143	Internal error applying the requested license
1115152	License does not apply to the platform
1115154	Unable to retrieve cluster ID
1115155	Invalid cluster ID found
1115159	License is not in an acceptable format
1115164	Minimum ONTAP version requirements not met
1115179	FlexCache is not supported on this system
1115180	FlexCache is not supported on cloud systems
1115407	Capacity pool licenses cannot be installed directly
66846818	Failed to interpret FlexCache license information
66846821	FlexCache is not supported on cloud systems
66846822	Invalid FlexCache capacity information provided
655294464	Failed to extract license contents
655294465	License key is invalid
655294466	Serial number is invalid
655294467	Version number is invalid
655294468	Expired license
655294469	License does not apply to the platform
655294470	License does not apply to the product

Name	Type	Description
errors	array[error]	

Example error

```
{
  "errors": [
    {
      "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	

capacity

Name	Type	Description
maximum_size	integer	Licensed capacity size (in bytes) that can be used.
used_size	integer	Capacity that is currently used (in bytes).

compliance

Name	Type	Description
state	string	Compliance state of the license.

licenses

Name	Type	Description
active	boolean	A flag indicating whether the license is currently being enforced.
capacity	capacity	
compliance	compliance	
evaluation	boolean	A flag indicating whether the license is in evaluation mode.
expiry_time	string	Date and time when the license expires.
owner	string	Cluster, node or license manager that owns the license.

Name	Type	Description
serial_number	string	Serial number of the license.
start_time	string	Date and time when the license starts.

license_package

Name	Type	Description
_links	_links	
keys	array[string]	
licenses	array[licenses]	Installed licenses of the package.
name	string	Name of the license.
scope	string	Scope of the license.
state	string	Summary state of package based on all installed licenses.

_links

Name	Type	Description
next	href	
self	href	

records

Name	Type	Description
_links	_links	
keys	array[string]	
licenses	array[licenses]	Installed licenses of the package.
name	string	Name of the license.
scope	string	Scope of the license.
state	string	Summary state of package based on all installed licenses.

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