



Manage key managers

ONTAP 9.8 REST API reference

NetApp
August 29, 2024

Table of Contents

- Manage key managers 1
 - Security key-managers endpoint overview 1
 - Retrieve key managers 19
 - Create a key manager 33
 - Delete key managers 48
 - Retrieve key managers 51
 - Update key managers 61

Manage key managers

Security key-managers endpoint overview

Overview

A key manager is a key management solution (software or dedicated hardware) that enables other ONTAP client modules to securely and persistently store keys for various uses. For example, WAFL uses the key management framework to store and retrieve the volume encryption keys that it uses to encrypt/decrypt data on NVE volumes. A key manager can be configured at both cluster scope and SVM, with one key manager allowed per SVM. The key management framework in ONTAP supports two mutually exclusive modes for persisting keys: external and onboard.

When an SVM is configured with external key management, the keys are stored on up to four key servers that are external to the system.

Once external key management is enabled for an SVM, key servers can be added or removed using the `/api/security/key-managers/{uuid}/key-servers` endpoint. See [POST `/security/key-managers/{uuid}/key-servers`] and [DELETE `/security/key-managers/{uuid}/key-servers/{server}`] for more details.

Setting up external key management dictates that the required certificates for securely communicating with the key server are installed prior to configuring the key manager. To install the required client and server_ca certificates, use the `/api/security/certificates/` endpoint.

See [POST `/security/certificates`], [GET `/security/certificates/uuid`] and [DELETE `/security/certificates/{uuid}`] for more details.

When an SVM is configured with the Onboard Key Manager, the keys are stored in ONTAP in wrapped format using a key hierarchy created using the salted hash of the passphrase entered when configuring the Onboard Key Manager. This model fits well for customers who use ONTAP to store their own data.

Examples

Creating an external key manager with 1 key server for a cluster

The example key manager is configured at the cluster-scope with one key server. Note that the UUIDs of the certificates are those that are already installed at the cluster-scope. Note the `return_records=true` query parameter is used to obtain the newly created key manager configuration

```
# The API:
POST /api/security/key-managers

# The call:
curl -X POST 'https://<mgmt-ip>/api/security/key-
managers?return_records=true' -H 'accept: application/hal+json' -d "{
\"external\": { \"client_certificate\": { \"uuid\": \"5fb1701a-d922-11e8-
bfe8-005056bb017d\" }, \"server_ca_certificates\": [ { \"uuid\":
\"827d7d31-d6c8-11e8-b5bf-005056bb017d\" } ],\"servers\": [ { \"server\":
\"10.225.89.33:5696\" } ] } }"

# The response:
{
  "num_records": 1,
  "records": [
    {
      "uuid": "815e9462-dc57-11e8-9b2c-005056bb017d",
      "external": {
        "client_certificate": {
          "uuid": "5fb1701a-d922-11e8-bfe8-005056bb017d"
        },
        "server_ca_certificates": [
          {
            "uuid": "827d7d31-d6c8-11e8-b5bf-005056bb017d"
          }
        ],
        "servers": [
          {
            "server": "10.225.89.33:5696"
          }
        ]
      },
      "_links": {
        "self": {
          "href": "/api/security/key-managers/815e9462-dc57-11e8-9b2c-
005056bb017d"
        }
      }
    }
  ]
}
```

Creating an external key manager with 1 primary key server and 2 secondary key servers for a cluster

The example key manager is configured at the cluster-scope with one key server and two secondary key servers. Note that the UUIDs of the certificates are those that are already installed at the cluster-scope. Note the `return_records=true` query parameter is used to obtain the newly created key manager configuration

```
# The API:
POST /api/security/key-managers

# The call:
curl -X POST 'https://<mgmt-ip>/api/security/key-
managers?return_records=true' -H 'accept: application/hal+json' -d "{
\"external\": { \"client_certificate\": { \"uuid\": \"5fb1701a-d922-11e8-
bfe8-005056bb017d\" }, \"server_ca_certificates\": [ { \"uuid\":
\"827d7d31-d6c8-11e8-b5bf-005056bb017d\" }],\"servers\": [ { \"server\":
\"10.225.89.33:5696\", \"secondary_key_servers\": [ \"1.1.1.1\",
\"10.72.204.27:5696\" ] } ] } }"

# The response:
{
  "num_records": 1,
  "records": [
    {
      "uuid": "815e9462-dc57-11e8-9b2c-005056bb017d",
      "external": {
        "client_certificate": {
          "uuid": "5fb1701a-d922-11e8-bfe8-005056bb017d"
        },
        "server_ca_certificates": [
          {
            "uuid": "827d7d31-d6c8-11e8-b5bf-005056bb017d"
          }
        ],
        "servers": [
          {
            "server": "10.225.89.33:5696",
            "secondary_key_servers": [
              "1.1.1.1",
              "10.72.204.27:5096"
            ]
          }
        ]
      },
      "_links": {
        "self": {
          "href": "/api/security/key-managers/815e9462-dc57-11e8-9b2c-
005056bb017d"
        }
      }
    }
  ]
}
```

```
    }  
  }  
}  
]  
}
```

Creating an external key manager with 1 key server for an SVM

The example key manager is configured at the SVM-scope with one key server. Note that the UUIDs of the certificates are those that are already installed in that SVM. Note the *return_records=true* query parameter is used to obtain the newly created key manager configuration

```

# The API:
POST /api/security/key-managers

# The call:
curl -X POST 'https://<mgmt-ip>/api/security/key-
managers?return_records=true' -H 'accept: application/hal+json' -d "{
\"svm\": { \"uuid\": \"216e6c26-d6c6-11e8-b5bf-005056bb017d\" },
\"external\": { \"client_certificate\": { \"uuid\": \"91dcaf7c-dbbd-11e8-
9b2c-005056bb017d\" }, \"server_ca_certificates\": [ { \"uuid\":
\"a4d4b8ba-dbbd-11e8-9b2c-005056bb017d\" } ], \"servers\": [ { \"server\":
\"10.225.89.34:5696\" } ] } }"

# The response:
{
  "num_records": 1,
  "records": [
    {
      "uuid": "80af63f2-dbbf-11e8-9b2c-005056bb017d",
      "svm": {
        "uuid": "216e6c26-d6c6-11e8-b5bf-005056bb017d"
      },
      "external": {
        "client_certificate": {
          "uuid": "91dcaf7c-dbbd-11e8-9b2c-005056bb017d"
        },
        "server_ca_certificates": [
          {
            "uuid": "a4d4b8ba-dbbd-11e8-9b2c-005056bb017d"
          }
        ],
        "servers": [
          {
            "server": "10.225.89.34:5696"
          }
        ]
      },
      "_links": {
        "self": {
          "href": "/api/security/key-managers/80af63f2-dbbf-11e8-9b2c-
005056bb017d"
        }
      }
    }
  ]
}

```

Creating an onboard key manager for a cluster

The following example shows how to create an onboard key manager for a cluster with the onboard key manager configured at the cluster-scope.

```
# The API:
POST /api/security/key-managers

# The call:
curl -X POST 'https://<mgmt-ip>/api/security/key-managers' -H 'accept:
application/hal+json' -d '{ "onboard": { "passphrase": "passphrase" } }'
```

Retrieving the key manager configurations for all clusters and SVMs

The following example shows how to retrieve all configured key managers along with their configurations.

```
# The API:
GET /api/security/key-managers

# The call:
curl -X GET 'https://<mgmt-ip>/api/security/key-managers?fields=*' -H
'accept: application/hal+json'

# The response:
{
  "records": [
    {
      "uuid": "2345f09c-d6c9-11e8-b5bf-005056bb017d",
      "scope": "svm",
      "svm": {
        "uuid": "0f22f8f3-d6c6-11e8-b5bf-005056bb017d",
        "name": "vs0"
      },
      "external": {
        "client_certificate": {
          "uuid": "4cb15482-d6c8-11e8-b5bf-005056bb017d",
          "_links": {
            "self": {
              "href": "/api/security/certificates/4cb15482-d6c8-11e8-b5bf-
005056bb017d/"
            }
          }
        }
      },
      "server_ca_certificates": [
```



```

    {
      "uuid": "8a17c858-d6c8-11e8-b5bf-005056bb017d",
      "_links": {
        "self": {
          "href": "/api/security/certificates/8a17c858-d6c8-11e8-b5bf-005056bb017d/"
        }
      }
    }
  ],
  "servers": [
    {
      "server": "10.2.30.4:5696",
      "timeout": 25,
      "username": "",
      "_links": {
        "self": {
          "href": "/api/security/key-managers/2345f09c-d6c9-11e8-b5bf-005056bb017d/key-servers/10.2.30.4:5696/"
        }
      }
    },
    {
      "server": "vs0.local1:3678",
      "timeout": 25,
      "username": "",
      "secondary_key_servers": [
        "1.1.1.1",
        "10.72.204.27:5096"
      ],
      "_links": {
        "self": {
          "href": "/api/security/key-managers/2345f09c-d6c9-11e8-b5bf-005056bb017d/key-servers/vs0.local1:3678/"
        }
      }
    }
  ]
},
"_links": {
  "self": {
    "href": "/api/security/key-managers/2345f09c-d6c9-11e8-b5bf-005056bb017d"
  }
}
},

```

```

{
  "uuid": "815e9462-dc57-11e8-9b2c-005056bb017d",
  "scope": "cluster",
  "external": {
    "client_certificate": {
      "uuid": "5fb1701a-d922-11e8-bfe8-005056bb017d",
      "_links": {
        "self": {
          "href": "/api/security/certificates/5fb1701a-d922-11e8-bfe8-005056bb017d/"
        }
      }
    },
    "server_ca_certificates": [
      {
        "uuid": "827d7d31-d6c8-11e8-b5bf-005056bb017d",
        "_links": {
          "self": {
            "href": "/api/security/certificates/827d7d31-d6c8-11e8-b5bf-005056bb017d/"
          }
        }
      }
    ],
    "servers": [
      {
        "server": "10.225.89.33:5696",
        "timeout": 25,
        "username": "",
        "_links": {
          "self": {
            "href": "/api/security/key-managers/815e9462-dc57-11e8-9b2c-005056bb017d/key-servers/10.225.89.33:5696/"
          }
        }
      }
    ]
  },
  "_links": {
    "self": {
      "href": "/api/security/key-managers/815e9462-dc57-11e8-9b2c-005056bb017d"
    }
  }
},

```

```

"num_records": 2,
"_links": {
  "self": {
    "href": "/api/security/key-managers?fields=*"
  }
}
}
}

```

Retrieving the key manager configurations for all clusters and SVMs (showing Onboard Key Manager)

The following example shows how to retrieve all configured key managers along with their configurations.

```

# The API:
GET /api/security/key-managers

# The call:
curl -X GET 'https://<mgmt-ip>/api/security/key-managers?fields=*' -H
'accept: application/hal+json'

# The response:
{
  "records": [
    {
      "uuid": "8ba52e0f-ae22-11e9-b747-005056bb7636",
      "scope": "cluster",
      "onboard": {
        "enabled": true,
        "key_backup": "-----BEGIN
BACKUP-----\n <Backup Data>
\n-----END BACKUP-----\n"
      },
      "volume_encryption": {
        "supported": false,
        "message": "The following nodes do not support volume granular
encryption: ntap-vsrm2.",
        "code": 65536935
      },
      "is_default_data_at_rest_encryption_disabled": false
    }
  ],
  "num_records": 1
}

```

Retrieving expensive fields such as, status.code and status.message, associated with a key manager.

These values are not retrieved by default with the 'fields=*' option. The following example shows how to retrieve the expensive objects associated with a key manager.

```
# The API:
GET /api/security/key-managers

# The call:
curl -X GET "https://<mgmt-ip>/api/security/key-managers?fields=status.message,status.code" -H 'accept: application/hal+json'

# The response:
{
  "records": [
    {
      "uuid": "ac305d46-aef4-11e9-ad3c-005056bb7636",
      "status": {
        "message": "No action needed at this time.",
        "code": 65537200
      },
      "_links": {
        "self": {
          "href": "/api/security/key-managers/ac305d46-aef4-11e9-ad3c-005056bb7636"
        }
      }
    }
  ],
  "num_records": 1,
  "_links": {
    "self": {
      "href": "/api/security/key-managers?fields=status.message,status.code"
    }
  }
}
```

Retrieving a specific key manager configuration

The following example shows how to retrieve a specific key manager configuration.

```
# The API:
GET /api/security/key-managers/{uuid}
```

```

# The call:
curl -X GET 'https://<mgmt-ip>/api/security/key-managers/<uuid>?fields=*'
-H 'accept: application/hal+json'

# The response:
{
  "uuid": "2345f09c-d6c9-11e8-b5bf-005056bb017d",
  "scope": "svm",
  "svm": {
    "uuid": "0f22f8f3-d6c6-11e8-b5bf-005056bb017d",
    "name": "vs0"
  },
  "external": {
    "client_certificate": {
      "uuid": "4cb15482-d6c8-11e8-b5bf-005056bb017d",
      "_links": {
        "self": {
          "href": "/api/security/certificates/4cb15482-d6c8-11e8-b5bf-005056bb017d/"
        }
      }
    },
    "server_ca_certificates": [
      {
        "uuid": "8a17c858-d6c8-11e8-b5bf-005056bb017d",
        "_links": {
          "self": {
            "href": "/api/security/certificates/8a17c858-d6c8-11e8-b5bf-005056bb017d/"
          }
        }
      }
    ],
    "servers": [
      {
        "server": "10.2.30.4:5696",
        "timeout": 25,
        "username": "",
        "_links": {
          "self": {
            "href": "/api/security/key-managers/2345f09c-d6c9-11e8-b5bf-005056bb017d/key-servers/10.2.30.4:5696/"
          }
        }
      }
    ],
  },
}

```

```

    {
      "server": "vs0.local1:3678",
      "timeout": 25,
      "username": "",
      "_links": {
        "self": {
          "href": "/api/security/key-managers/2345f09c-d6c9-11e8-b5bf-
005056bb017d/key-servers/vs0.local1:3678/"
        }
      }
    }
  ],
  "_links": {
    "self": {
      "href": "/api/security/key-managers/2345f09c-d6c9-11e8-b5bf-
005056bb017d"
    }
  }
}

```

Updating the configuration of an external key manager

The following example shows how to update the `server_ca` configuration of an external key manager.

```

# The API:
PATCH /api/security/key-managers/{uuid}

# The call:
curl -X PATCH 'https://<mgmt-ip>/api/security/key-managers/<uuid>?' -H
'accept: application/hal+json' -d "{ \"external\": {
  \"server_ca_certificates\": [ { \"uuid\": \"23b05c58-d790-11e8-b5bf-
005056bb017d\" } ] } }"

```

Updating the passphrase of an Onboard Key Manager

The following example shows how to update the passphrase of a given key manager.

```
# The API:
PATCH /api/security/key-managers/{uuid}

# The call:
curl -X PATCH 'https://<mgmt-ip>/api/security/key-managers/<uuid>?' -H
'accept: application/hal+json' -d "{ \"onboard\": {
\"existing_passphrase\": \"existing_passphrase\", \"passphrase\":
\"new_passphrase\" } }"
```

Synchronizing the passphrase of the Onboard Key Manager on a cluster

The following example shows how to synchronize the passphrase on a cluster where the Onboard Key Manager is already configured.

```
# The API:
PATCH /api/security/key-managers/{uuid}

# The call:
curl -X PATCH 'https://<mgmt-ip>/api/security/key-managers/<uuid>?' -H
'accept: application/hal+json' -d "{ \"onboard\": {
\"existing_passphrase\": \"existing_passphrase\", \"synchronize\": true
}}"
```

Configuring the Onboard Key Manager on a cluster

The following example shows how to configure the Onboard Key Manager on a cluster where the Onboard Key Manager is not configured, but is configured on an MetroCluster partner cluster.

```
# The API:
POST /api/security/key-managers

# The call:
curl -X POST 'https://<mgmt-ip>/api/security/key-
managers?return_records=false' -H 'accept: application/hal+json' -H
"Content-Type: application/json" -d "{ \"onboard\": { \"passphrase\":
\"passphrase\", \"synchronize\": true } }"
```

Deleting a configured key manager

The following example shows how to delete a key manager given its UUID.

```
# The API:  
DELETE /api/security/key-managers/{uuid}  
  
# The call:  
curl -X DELETE 'https://<mgmt-ip>/api/security/key-managers/<uuid>?' -H  
'accept: application/hal+json'
```

Adding a key server to an external key manager

The following example shows how to add a key server with two secondary key servers to an external key manager.


```

# The API:
POST /api/security/key-managers/{uuid}/key-servers

# The call:
curl -X POST 'https://<mgmt-ip>/api/security/key-managers/<uuid>/key-
servers?return_records=true' -H 'accept: application/hal+json' -d "{
  \"server\": \"10.225.89.34:5696\", \"secondary_key_servers\": [
  \"1.1.1.1\", \"10.72.204.27:5696\" ] }"

# The response:
{
  "num_records": 1,
  "records": [
    {
      "server": "10.225.89.34:5696",
      "secondary_key_servers": [
        "1.1.1.1",
        "10.72.204.27:5096"
      ],
      "_links": {
        "self": {
          "href": "/api/security/key-managers/43e0c191-dc5c-11e8-9b2c-
005056bb017d/key-servers/10.225.89.34%3A5696"
        }
      }
    }
  ]
}

```

Adding 2 key servers to an external key manager

The following example shows how to add 2 key servers to an external key manager. Note that the *records* property is used to add multiple key servers to the key manager in a single API call.

```
# The API:
POST /api/security/key-managers/{uuid}/key-servers

# The call:
curl -X POST 'https://<mgmt-ip>/api/security/key-managers/<uuid>/key-servers?return_records=true' -H 'accept: application/hal+json' -d '{"records": [ { "server": "10.225.89.34:5696" }, { "server": "10.225.89.33:5696" } ] }'

# The response:
{
  "num_records": 1,
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/security/key-managers/43e0c191-dc5c-11e8-9b2c-005056bb017d/key-servers/"
        }
      }
    }
  ]
}
```

Retrieving all the key servers configured in an external key manager

The following example shows how to retrieve all key servers configured in an external key manager.

```
# The API:
GET /api/security/key-managers/{uuid}/key-servers

# The call:
curl -X GET 'https://<mgmt-ip>/api/security/key-managers/<uuid>/key-servers?fields=*' -H 'accept: application/hal+json'

# The response:
{
  "records": [
    {
      "uuid": "43e0c191-dc5c-11e8-9b2c-005056bb017d",
      "server": "10.225.89.33:5696",
      "timeout": 25,
      "username": ""
    }
  ]
}
```

```

    "secondary_key_servers": [
      "1.1.1.1",
      "10.72.204.27:5096"
    ],
    "_links": {
      "self": {
        "href": "/api/security/key-managers/43e0c191-dc5c-11e8-9b2c-005056bb017d/key-servers/10.225.89.33%3A5696"
      }
    }
  },
  {
    "uuid": "43e0c191-dc5c-11e8-9b2c-005056bb017d",
    "server": "10.225.89.34:5696",
    "timeout": 25,
    "username": "",
    "_links": {
      "self": {
        "href": "/api/security/key-managers/43e0c191-dc5c-11e8-9b2c-005056bb017d/key-servers/10.225.89.34%3A5696"
      }
    }
  }
],
"num_records": 2,
"_links": {
  "self": {
    "href": "/api/security/key-managers/43e0c191-dc5c-11e8-9b2c-005056bb017d/key-servers?fields=*"
  }
}
}

```

Retrieving a specific key server configured in an external key manager

The following example shows how to retrieve a specific key server configured in an external key manager.

```
# The API:
GET /api/security/key-managers/{uuid}/key-servers/{server}

# The call:
curl -X GET 'https://<mgmt-ip>/api/security/key-managers/<uuid>/key-servers/{server}?fields=*' -H 'accept: application/hal+json'

# The response:
{
  "uuid": "43e0c191-dc5c-11e8-9b2c-005056bb017d",
  "server": "10.225.89.34:5696",
  "timeout": 25,
  "username": "",
  "_links": {
    "self": {
      "href": "/api/security/key-managers/43e0c191-dc5c-11e8-9b2c-005056bb017d/key-servers/10.225.89.34:5696"
    }
  }
}
```

Updating a specific key server configuration configured in an external key manager

The following example shows how to update a specific key server configured in an external key manager.

```
# The API:
PATCH /api/security/key-managers/{uuid}/key-servers/{server}

# The call:
curl -X PATCH 'https://<mgmt-ip>/api/security/key-managers/<uuid>/key-servers/{server}' -H 'accept: application/hal+json' -d '{"timeout": 45}'
```

The following example shows how to update the set of secondary key servers associated with a key server.

```
# The API:
PATCH /api/security/key-managers/{uuid}/key-servers/{server}

# The call:
curl -X PATCH 'https://<mgmt-ip>/api/security/key-managers/<uuid>/key-servers/{server}' -H 'accept: application/hal+json' -d "{
  \"secondary_key_servers\": [ \"1.1.1.1\", \"10.72.204.27:5696\" ] }"
```

Deleting a key server from an external key manager

The following example shows how to delete a key server from an external key manager.

```
# The API:
DELETE /api/security/key-managers/{uuid}/key-servers/{server}

# The call:
curl -X DELETE 'https://<mgmt-ip>/api/security/key-managers/<uuid>/key-servers/{server}' -H 'accept: application/hal+json'
```

Retrieve key managers

GET /security/key-managers

Introduced In: 9.6

Retrieves key managers.

Expensive properties

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

- `status.message`
- `status.code`

Related ONTAP commands

- `security key-manager show-keystore`
- `security key-manager external show`
- `security key-manager external show-status`

- security key-manager onboard show-backup

Parameters

Name	Type	In	Required	Description
uuid	string	query	False	Filter by uuid
scope	string	query	False	Filter by scope
volume_encryption.code	integer	query	False	Filter by volume_encryption.code • Introduced in: 9.7
volume_encryption.supported	boolean	query	False	Filter by volume_encryption.supported • Introduced in: 9.7
volume_encryption.message	string	query	False	Filter by volume_encryption.message • Introduced in: 9.7
onboard.key_backup	string	query	False	Filter by onboard.key_backup • Introduced in: 9.7
onboard.enabled	boolean	query	False	Filter by onboard.enabled
is_default_data_at_rest_encryption_disabled	boolean	query	False	Filter by is_default_data_at_rest_encryption_disabled • Introduced in: 9.7
svm.uuid	string	query	False	Filter by svm.uuid

Name	Type	In	Required	Description
svm.name	string	query	False	Filter by svm.name
status.code	integer	query	False	Filter by status.code • Introduced in: 9.7
status.message	string	query	False	Filter by status.message • Introduced in: 9.7
external.server_ca_certificates.name	string	query	False	Filter by external.server_ca_certificates.name • Introduced in: 9.8
external.server_ca_certificates.uuid	string	query	False	Filter by external.server_ca_certificates.uuid
external.client_certificate.name	string	query	False	Filter by external.client_certificate.name • Introduced in: 9.8
external.client_certificate.uuid	string	query	False	Filter by external.client_certificate.uuid
external.servers.server	string	query	False	Filter by external.servers.server
external.servers.username	string	query	False	Filter by external.servers.username

Name	Type	In	Required	Description
external.servers.secondary_key_servers	string	query	False	Filter by external.servers.secondary_key_servers <ul style="list-style-type: none"> • Introduced in: 9.8
external.servers.timeout	integer	query	False	Filter by external.servers.timeout
fields	array[string]	query	False	Specify the fields to return.
max_records	integer	query	False	Limit the number of records returned.
return_timeout	integer	query	False	The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached. <ul style="list-style-type: none"> • Default value: 1 • Max value: 120 • Min value: 0
return_records	boolean	query	False	The default is true for GET calls. When set to false, only the number of records is returned. <ul style="list-style-type: none"> • Default value: 1
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[security_key_manager]	

Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "external": {
        "client_certificate": {
          "_links": {
            "self": {
              "href": "/api/resourcelink"
            }
          },
          "name": "cert1",
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
        },
        "server_ca_certificates": [
          {
            "_links": {
              "self": {
                "href": "/api/resourcelink"
              }
            },
            "name": "cert1",
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
          }
        ],
        "servers": [
          {
            "_links": {
              "self": {
                "href": "/api/resourcelink"
              }
            }
          }
        ]
      }
    }
  ]
}
```



```

"scope": "string",
"status": {
  "code": "346758",
  "message": "This cluster is part of a MetroCluster
configuration. Use the REST API POST method security/key_managers/ with
the synchronize option and the same passphrase on the partner cluster
before proceeding with any key manager operations. Failure to do so
could lead to switchover or switchback failure."
},
"svm": {
  "_links": {
    "self": {
      "href": "/api/resource/link"
    }
  },
  "name": "svm1",
  "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
},
"uuid": "string",
"volume_encryption": {
  "code": "346758",
  "message": "No platform support for volume encryption in
following nodes - node1, node2."
}
}
]
}

```

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	

client_certificate

Client certificate

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

server_ca_certificates

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

self_link

Name	Type	Description
self	href	

key_server_readcreate

Name	Type	Description
_links	self_link	
secondary_key_servers	array[string]	A list of the secondary key servers associated with the primary key server.
server	string	External key server for key management. If no port is provided, a default port of 5696 is used.
timeout	integer	I/O timeout in seconds for communicating with the key server.
username	string	Username credentials for connecting with the key server.

external

Configures external key management

Name	Type	Description
client_certificate	client_certificate	Client certificate
server_ca_certificates	array[server_ca_certificates]	The UUIDs of the server CA certificates already installed in the cluster or SVM. The array of certificates are common for all the key servers per SVM.
servers	array[key_server_readcreate]	The set of external key servers.

onboard

Configures onboard key management. After configuring onboard key management, save the encrypted configuration data in a safe location so that you can use it if you need to perform a manual recovery operation.

Name	Type	Description
enabled	boolean	Is the onboard key manager enabled?
existing_passphrase	string	The cluster-wide passphrase. This is not audited.

Name	Type	Description
key_backup	string	Backup of the onboard key manager's key hierarchy. It is required to save this backup after configuring the onboard key manager to help in the recovery of the cluster in case of catastrophic failures.
passphrase	string	The cluster-wide passphrase. This is not audited.
synchronize	boolean	Synchronizes missing onboard keys on any node in the cluster. If a node is added to a cluster that has onboard key management configured, the synchronize operation needs to be performed in a PATCH operation. In a MetroCluster configuration, if onboard key management is enabled on one site, then the synchronize operation needs to be run as a POST operation on the remote site providing the same passphrase.

status

Optional status information on the current state of the key manager indicating if it is fully setup or requires more action.

Name	Type	Description
code	integer	Code corresponding to the status message. Returns 0 if the setup is complete. This is an advanced property; there is an added cost to retrieving its value. The property is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter or GET for all advanced properties is enabled.

Name	Type	Description
message	string	Current state of the key manager indicating any additional steps to perform to finish the setup. This is an advanced property; there is an added cost to retrieving its value. The property is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter or GET for all advanced properties is enabled.

svm

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

volume_encryption

Indicates whether volume encryption is supported in the cluster.

Name	Type	Description
code	integer	Code corresponding to the status message. Returns a 0 if volume encryption is supported in all nodes of the cluster.
message	string	Reason for not supporting volume encryption.
supported	boolean	Set to true when volume encryption support is available on all nodes of the cluster.

security_key_manager

Name	Type	Description
_links	_links	

Name	Type	Description
external	external	Configures external key management
is_default_data_at_rest_encryption_disabled	boolean	Indicates whether default data-at-rest encryption is disabled in the cluster.
onboard	onboard	Configures onboard key management. After configuring onboard key management, save the encrypted configuration data in a safe location so that you can use it if you need to perform a manual recovery operation.
scope	string	Set to "svm" for interfaces owned by an SVM. Otherwise, set to "cluster".
status	status	Optional status information on the current state of the key manager indicating if it is fully setup or requires more action.
svm	svm	
uuid	string	
volume_encryption	volume_encryption	Indicates whether volume encryption is supported in the cluster.

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

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

Create a key manager

POST /security/key-managers

Introduced In: 9.6

Creates a key manager.

Required properties

- `svm.uuid` or `svm.name` - Existing SVM in which to create a key manager.
- `external.client_certificate` - Client certificate. Required only when creating an external key manager.
- `external.server_ca_certificates` - Server CA certificates. Required only when creating an external key manager.
- `external.servers.server` - Key servers. Required only when creating an external key manager.
- `onboard.passphrase` - Cluster-wide passphrase. Required only when creating an Onboard Key Manager.
- `synchronize` - Synchronizes missing onboard keys on any node in the cluster. Required only when creating an Onboard Key Manager at the partner site of a MetroCluster configuration.

Related ONTAP commands

- `security key-manager external enable`
- `security key-manager onboard enable`
- `security key-manager onboard sync`

Parameters

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

Request Body

Name	Type	Description
_links	_links	
external	external	Configures external key management
is_default_data_at_rest_encryption_disabled	boolean	Indicates whether default data-at-rest encryption is disabled in the cluster.
onboard	onboard	Configures onboard key management. After configuring onboard key management, save the encrypted configuration data in a safe location so that you can use it if you need to perform a manual recovery operation.
scope	string	Set to "svm" for interfaces owned by an SVM. Otherwise, set to "cluster".
status	status	Optional status information on the current state of the key manager indicating if it is fully setup or requires more action.
svm	svm	
uuid	string	
volume_encryption	volume_encryption	Indicates whether volume encryption is supported in the cluster.

Example request

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "external": {
    "client_certificate": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "cert1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "server_ca_certificates": [
      {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "cert1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      }
    ],
    "servers": [
      {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "secondary_key_servers": [
          "string"
        ],
        "server": "keyserver1.com:5698",
        "timeout": "60",
        "username": "admin"
      }
    ]
  },
  "onboard": {
```



```
"_links": {
  "self": {
    "href": "/api/resourcelink"
  }
},
"name": "svm1",
"uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
},
"uuid": "string",
"volume_encryption": {
  "code": "346758",
  "message": "No platform support for volume encryption in following
nodes - node1, node2."
}
}
```

Response

Status: 201, Created

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

Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "records": [
    {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "external": {
        "client_certificate": {
          "_links": {
            "self": {
              "href": "/api/resourcelink"
            }
          },
          "name": "cert1",
          "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
        },
        "server_ca_certificates": [
          {
            "_links": {
              "self": {
                "href": "/api/resourcelink"
              }
            },
            "name": "cert1",
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
          }
        ],
        "servers": [
          {
            "_links": {
              "self": {
                "href": "/api/resourcelink"
              }
            }
          }
        ]
      }
    }
  ]
}
```



```

"scope": "string",
"status": {
  "code": "346758",
  "message": "This cluster is part of a MetroCluster
configuration. Use the REST API POST method security/key_managers/ with
the synchronize option and the same passphrase on the partner cluster
before proceeding with any key manager operations. Failure to do so
could lead to switchover or switchback failure."
},
"svm": {
  "_links": {
    "self": {
      "href": "/api/resource/link"
    }
  },
  "name": "svm1",
  "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
},
"uuid": "string",
"volume_encryption": {
  "code": "346758",
  "message": "No platform support for volume encryption in
following nodes - node1, node2."
}
}
]
}

```

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
65536038	A maximum of 4 active key servers are allowed.
65536214	Failed to generate cluster key encryption key.
65536216	Failed to add cluster key encryption key.
65536310	Failed to setup the Onboard Key Manager because the MetroCluster peer is unhealthy.
65536508	The platform does not support data at rest encryption.
65536821	The certificate is not installed.

Error Code	Description
65536822	Multitenant key management is not supported in the current cluster version.
65536823	The SVM has key manager already configured.
65536824	Multitenant key management is not supported in MetroCluster configurations.
65536834	Failed to get existing key-server details for the SVM.
65536852	Failed to query supported KMIP protocol versions.
65536870	Key management servers already configured.
65536871	Duplicate key management servers exist.
65536876	External key management requires client and server CA certificates installed and with one or more key servers provided.
65536878	External key management cannot be configured as one or more volume encryption keys of the SVM are stored in cluster key management server.
65536895	External key manager cannot be configured since this cluster is part of a MetroCluster configuration and the partner site of this MetroCluster configuration has Onboard Key Manager configured.
65536900	The Onboard Key Manager cannot be configured because this cluster is part of a MetroCluster configuration and the partner site has the external key manager configured.
65536903	The Onboard Key Manager has failed to configure on some nodes in the cluster. Use the CLI to sync the Onboard Key Manager configuration on failed nodes.
65536906	The Onboard Key Manager has already been configured at the partner site. Use the CLI to sync the Onboard Key Manager with the same passphrase.
65536907	The Onboard Key Manager is already configured. Use the CLI to sync any nodes with the Onboard Key Manager configuration.
65536916	The Onboard Key Manager is only supported for an admin SVM.
65536920	The Onboard Key Manager passphrase length is incorrect.
65537240	The Onboard Key Manager passphrase must be provided when performing a POST/synchronize operation.
65537241	The Onboard Key Manager existing_passphrase must not be provided when performing a POST/synchronize operation.

Error Code	Description
65537244	Unable to sync/create Onboard Key Manager on the local cluster; Onboard Key Manager is already configured on the cluster.
65537245	Unable to sync/create Onboard Key Manager on the local cluster; Onboard Key Manager is not configured on the partner cluster.
65537246	Unable to sync/create Onboard Key Manager on local cluster. This cluster is not part of a MetroCluster configuration.
66060338	Failed to establish secure connection for a key management server due to incorrect server_ca certificates.
66060339	Failed to establish secure connection for a key management server due to incorrect client certificates.
66060340	Failed to establish secure connection for a key management server due to Cryptsoft 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	

client_certificate

Client certificate

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

server_ca_certificates

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

self_link

Name	Type	Description
self	href	

key_server_readcreate

Name	Type	Description
_links	self_link	
secondary_key_servers	array[string]	A list of the secondary key servers associated with the primary key server.

Name	Type	Description
server	string	External key server for key management. If no port is provided, a default port of 5696 is used.
timeout	integer	I/O timeout in seconds for communicating with the key server.
username	string	Username credentials for connecting with the key server.

external

Configures external key management

Name	Type	Description
client_certificate	client_certificate	Client certificate
server_ca_certificates	array[server_ca_certificates]	The UUIDs of the server CA certificates already installed in the cluster or SVM. The array of certificates are common for all the key servers per SVM.
servers	array[key_server_readcreate]	The set of external key servers.

onboard

Configures onboard key management. After configuring onboard key management, save the encrypted configuration data in a safe location so that you can use it if you need to perform a manual recovery operation.

Name	Type	Description
enabled	boolean	Is the onboard key manager enabled?
existing_passphrase	string	The cluster-wide passphrase. This is not audited.

Name	Type	Description
key_backup	string	Backup of the onboard key manager's key hierarchy. It is required to save this backup after configuring the onboard key manager to help in the recovery of the cluster in case of catastrophic failures.
passphrase	string	The cluster-wide passphrase. This is not audited.
synchronize	boolean	Synchronizes missing onboard keys on any node in the cluster. If a node is added to a cluster that has onboard key management configured, the synchronize operation needs to be performed in a PATCH operation. In a MetroCluster configuration, if onboard key management is enabled on one site, then the synchronize operation needs to be run as a POST operation on the remote site providing the same passphrase.

status

Optional status information on the current state of the key manager indicating if it is fully setup or requires more action.

Name	Type	Description
code	integer	Code corresponding to the status message. Returns 0 if the setup is complete. This is an advanced property; there is an added cost to retrieving its value. The property is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter or GET for all advanced properties is enabled.

Name	Type	Description
message	string	Current state of the key manager indicating any additional steps to perform to finish the setup. This is an advanced property; there is an added cost to retrieving its value. The property is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter or GET for all advanced properties is enabled.

svm

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

volume_encryption

Indicates whether volume encryption is supported in the cluster.

Name	Type	Description
code	integer	Code corresponding to the status message. Returns a 0 if volume encryption is supported in all nodes of the cluster.
message	string	Reason for not supporting volume encryption.
supported	boolean	Set to true when volume encryption support is available on all nodes of the cluster.

security_key_manager

Name	Type	Description
_links	_links	

Name	Type	Description
external	external	Configures external key management
is_default_data_at_rest_encryption_disabled	boolean	Indicates whether default data-at-rest encryption is disabled in the cluster.
onboard	onboard	Configures onboard key management. After configuring onboard key management, save the encrypted configuration data in a safe location so that you can use it if you need to perform a manual recovery operation.
scope	string	Set to "svm" for interfaces owned by an SVM. Otherwise, set to "cluster".
status	status	Optional status information on the current state of the key manager indicating if it is fully setup or requires more action.
svm	svm	
uuid	string	
volume_encryption	volume_encryption	Indicates whether volume encryption is supported in the cluster.

_links

Name	Type	Description
next	href	
self	href	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Delete key managers

DELETE /security/key-managers/{uuid}

Introduced In: 9.6

Deletes a key manager.

Related ONTAP commands

- `security key-manager external disable`
- `security key-manager onboard disable`

Parameters

Name	Type	In	Required	Description
uuid	string	path	True	Key manager UUID

Response

Status: 200, Ok

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
65536208	Failed to delete the SVM Key ID.

Error Code	Description
65536233	Internal error. Deletion of km_wrapped_kdb key database has failed for the Onboard Key Manager.
65536234	Internal error. Deletion of cluster_kdb key database has failed for the Onboard Key Manager.
65536239	Encrypted volumes are found for the SVM.
65536242	One or more Storage Encryption devices are assigned an authentication key.
65536242	One or more Storage Encryption devices are assigned an authentication key.
65536800	Failed to lookup onboard keys.
65536813	Encrypted kernel core files found.
65536817	Failed to determine if key manager is safe to disable.
65536822	Multitenant key management is not supported in the current cluster version.
65536827	Failed to determine if the SVM has any encrypted volumes.
65536828	External key management is not enabled for the SVM.
65536867	Encrypted volumes are found for the SVM.
196608301	Failed to determine the type of encryption.
196608305	NAE aggregates are found in the cluster.

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Retrieve key managers

GET /security/key-managers/{uuid}

Introduced In: 9.6

Retrieves key managers.

Expensive properties

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

- `status.message`
- `status.code`

Related ONTAP commands

- `security key-manager show-keystore`
- `security key-manager external show`
- `security key-manager external show-status`
- `security key-manager onboard show-backup`

Parameters

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

Response

```
Status: 200, Ok
```

Name	Type	Description
<code>_links</code>	_links	
<code>external</code>	external	Configures external key management

Name	Type	Description
is_default_data_at_rest_encryption_disabled	boolean	Indicates whether default data-at-rest encryption is disabled in the cluster.
onboard	onboard	Configures onboard key management. After configuring onboard key management, save the encrypted configuration data in a safe location so that you can use it if you need to perform a manual recovery operation.
scope	string	Set to "svm" for interfaces owned by an SVM. Otherwise, set to "cluster".
status	status	Optional status information on the current state of the key manager indicating if it is fully setup or requires more action.
svm	svm	
uuid	string	
volume_encryption	volume_encryption	Indicates whether volume encryption is supported in the cluster.

Example response

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "external": {
    "client_certificate": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "cert1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "server_ca_certificates": [
      {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "cert1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      }
    ],
    "servers": [
      {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "secondary_key_servers": [
          "string"
        ],
        "server": "keyserver1.com:5698",
        "timeout": "60",
        "username": "admin"
      }
    ]
  },
  "onboard": {
```



```
"_links": {
  "self": {
    "href": "/api/resourcelink"
  }
},
"name": "svm1",
"uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
},
"uuid": "string",
"volume_encryption": {
  "code": "346758",
  "message": "No platform support for volume encryption in following
nodes - node1, node2."
}
}
```

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	

client_certificate

Client certificate

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

server_ca_certificates

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

self_link

Name	Type	Description
self	href	

key_server_readcreate

Name	Type	Description
_links	self_link	
secondary_key_servers	array[string]	A list of the secondary key servers associated with the primary key server.

Name	Type	Description
server	string	External key server for key management. If no port is provided, a default port of 5696 is used.
timeout	integer	I/O timeout in seconds for communicating with the key server.
username	string	Username credentials for connecting with the key server.

external

Configures external key management

Name	Type	Description
client_certificate	client_certificate	Client certificate
server_ca_certificates	array[server_ca_certificates]	The UUIDs of the server CA certificates already installed in the cluster or SVM. The array of certificates are common for all the key servers per SVM.
servers	array[key_server_readcreate]	The set of external key servers.

onboard

Configures onboard key management. After configuring onboard key management, save the encrypted configuration data in a safe location so that you can use it if you need to perform a manual recovery operation.

Name	Type	Description
enabled	boolean	Is the onboard key manager enabled?
existing_passphrase	string	The cluster-wide passphrase. This is not audited.

Name	Type	Description
key_backup	string	Backup of the onboard key manager's key hierarchy. It is required to save this backup after configuring the onboard key manager to help in the recovery of the cluster in case of catastrophic failures.
passphrase	string	The cluster-wide passphrase. This is not audited.
synchronize	boolean	Synchronizes missing onboard keys on any node in the cluster. If a node is added to a cluster that has onboard key management configured, the synchronize operation needs to be performed in a PATCH operation. In a MetroCluster configuration, if onboard key management is enabled on one site, then the synchronize operation needs to be run as a POST operation on the remote site providing the same passphrase.

status

Optional status information on the current state of the key manager indicating if it is fully setup or requires more action.

Name	Type	Description
code	integer	Code corresponding to the status message. Returns 0 if the setup is complete. This is an advanced property; there is an added cost to retrieving its value. The property is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter or GET for all advanced properties is enabled.

Name	Type	Description
message	string	Current state of the key manager indicating any additional steps to perform to finish the setup. This is an advanced property; there is an added cost to retrieving its value. The property is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter or GET for all advanced properties is enabled.

svm

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

volume_encryption

Indicates whether volume encryption is supported in the cluster.

Name	Type	Description
code	integer	Code corresponding to the status message. Returns a 0 if volume encryption is supported in all nodes of the cluster.
message	string	Reason for not supporting volume encryption.
supported	boolean	Set to true when volume encryption support is available on all nodes of the cluster.

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 key managers

PATCH /security/key-managers/{uuid}

Introduced In: 9.6

Updates a key manager.

Required properties

- `onboard.existing_passphrase` - Cluster-wide passphrase. Required only when synchronizing the passphrase of the Onboard Key Manager.
- `synchronize` - Synchronizes missing Onboard Key Manager keys on any node in the cluster. Required only when synchronizing the Onboard Key Manager keys in a local cluster.

Related ONTAP commands

- `security key-manager external modify`
- `security key-manager onboard sync`
- `security key-manager onboard update-passphrase`

Parameters

Name	Type	In	Required	Description
uuid	string	path	True	Key manager UUID

Request Body

Name	Type	Description
<code>_links</code>	_links	

Name	Type	Description
external	external	Configures external key management
is_default_data_at_rest_encryption_disabled	boolean	Indicates whether default data-at-rest encryption is disabled in the cluster.
onboard	onboard	Configures onboard key management. After configuring onboard key management, save the encrypted configuration data in a safe location so that you can use it if you need to perform a manual recovery operation.
scope	string	Set to "svm" for interfaces owned by an SVM. Otherwise, set to "cluster".
status	status	Optional status information on the current state of the key manager indicating if it is fully setup or requires more action.
svm	svm	
uuid	string	
volume_encryption	volume_encryption	Indicates whether volume encryption is supported in the cluster.

Example request

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "external": {
    "client_certificate": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "cert1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "server_ca_certificates": [
      {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "cert1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      }
    ],
    "servers": [
      {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "secondary_key_servers": [
          "string"
        ],
        "server": "keyserver1.com:5698",
        "timeout": "60",
        "username": "admin"
      }
    ]
  },
  "onboard": {
```



```

    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  },
  "uuid": "string",
  "volume_encryption": {
    "code": "346758",
    "message": "No platform support for volume encryption in following
nodes - node1, node2."
  }
}

```

Response

Status: 200, Ok

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
65536139	The existing passphrase value provided does not match the configured passphrase.
65536150	The new passphrase is same as old passphrase.
65536404	The passphrase does not match the accepted length.
65536406	The change of passphrase failed.
65536407	The passphrase update failed on some nodes.
65536408	The passphrase update failed on some nodes.
65536802	The passphrase does not match the accepted length in common criteria mode.
65536821	The certificate is not installed.
65536822	Multitenant key management is not supported in the current cluster version.

Error Code	Description
65536828	External key management is not enabled for the SVM.
65536850	New client certificate public or private keys are different from the existing client certificate.
65536852	Failed to query supported KMIP protocol versions.
65536917	Updating an onboard passphrase requires both new and existing cluster passphrase.
65537242	The Onboard Key Manager existing_passphrase must be provided when performing a PATCH/synchronize operation.
65537243	The Onboard Key Manager passphrase must not be provided when performing a PATCH/synchronize operation.
66060338	Failed to establish secure connection for a key management server due to incorrect server_ca certificates.
66060339	Failed to establish secure connection for a key management server due to incorrect client certificates.
66060340	Failed to establish secure connection for a key management server due to Cryptsoft 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	

client_certificate

Client certificate

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

server_ca_certificates

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

self_link

Name	Type	Description
self	href	

key_server_readcreate

Name	Type	Description
_links	self_link	
secondary_key_servers	array[string]	A list of the secondary key servers associated with the primary key server.

Name	Type	Description
server	string	External key server for key management. If no port is provided, a default port of 5696 is used.
timeout	integer	I/O timeout in seconds for communicating with the key server.
username	string	Username credentials for connecting with the key server.

external

Configures external key management

Name	Type	Description
client_certificate	client_certificate	Client certificate
server_ca_certificates	array[server_ca_certificates]	The UUIDs of the server CA certificates already installed in the cluster or SVM. The array of certificates are common for all the key servers per SVM.
servers	array[key_server_readcreate]	The set of external key servers.

onboard

Configures onboard key management. After configuring onboard key management, save the encrypted configuration data in a safe location so that you can use it if you need to perform a manual recovery operation.

Name	Type	Description
enabled	boolean	Is the onboard key manager enabled?
existing_passphrase	string	The cluster-wide passphrase. This is not audited.

Name	Type	Description
key_backup	string	Backup of the onboard key manager's key hierarchy. It is required to save this backup after configuring the onboard key manager to help in the recovery of the cluster in case of catastrophic failures.
passphrase	string	The cluster-wide passphrase. This is not audited.
synchronize	boolean	Synchronizes missing onboard keys on any node in the cluster. If a node is added to a cluster that has onboard key management configured, the synchronize operation needs to be performed in a PATCH operation. In a MetroCluster configuration, if onboard key management is enabled on one site, then the synchronize operation needs to be run as a POST operation on the remote site providing the same passphrase.

status

Optional status information on the current state of the key manager indicating if it is fully setup or requires more action.

Name	Type	Description
code	integer	Code corresponding to the status message. Returns 0 if the setup is complete. This is an advanced property; there is an added cost to retrieving its value. The property is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter or GET for all advanced properties is enabled.

Name	Type	Description
message	string	Current state of the key manager indicating any additional steps to perform to finish the setup. This is an advanced property; there is an added cost to retrieving its value. The property is not populated for either a collection GET or an instance GET unless it is explicitly requested using the <code>fields</code> query parameter or GET for all advanced properties is enabled.

svm

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

volume_encryption

Indicates whether volume encryption is supported in the cluster.

Name	Type	Description
code	integer	Code corresponding to the status message. Returns a 0 if volume encryption is supported in all nodes of the cluster.
message	string	Reason for not supporting volume encryption.
supported	boolean	Set to true when volume encryption support is available on all nodes of the cluster.

security_key_manager

Name	Type	Description
_links	_links	

Name	Type	Description
external	external	Configures external key management
is_default_data_at_rest_encryption_disabled	boolean	Indicates whether default data-at-rest encryption is disabled in the cluster.
onboard	onboard	Configures onboard key management. After configuring onboard key management, save the encrypted configuration data in a safe location so that you can use it if you need to perform a manual recovery operation.
scope	string	Set to "svm" for interfaces owned by an SVM. Otherwise, set to "cluster".
status	status	Optional status information on the current state of the key manager indicating if it is fully setup or requires more action.
svm	svm	
uuid	string	
volume_encryption	volume_encryption	Indicates whether volume encryption is supported in the cluster.

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

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

Copyright information

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.