Veritas NetBackup™ WebSocket Service (NBWSS) Reference Guide

Release 8.0

VERITAS™

Veritas NetBackup™ WebSocket Service (NBWSS) Reference Guide

Legal Notice

Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:

https://www.veritas.com/support

You can manage your Veritas account information at the following URL:

https://my.veritas.com

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

Worldwide (except Japan)	CustomerCare@veritas.com
Japan	CustomerCare_Japan@veritas.com

Documentation

Make sure that you have the current version of the documentation. Each document displays the date of the last update on page 2. The latest documentation is available on the Veritas website:

https://sort.veritas.com/documents

Documentation feedback

Your feedback is important to us. Suggest improvements or report errors or omissions to the documentation. Include the document title, document version, chapter title, and section title of the text on which you are reporting. Send feedback to:

NB.docs@veritas.com

You can also see documentation information or ask a question on the Veritas community site:

http://www.veritas.com/community/

Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

Contents

Using the NetBackup WebSocket Service (NBWSS) for communication with a cloud application

This chapter includes the following topics:

About the NetBackup WebSocket Service (NBWSS)

Veritas provides a NetBackup WebSocket Service (NBWSS) that allows applications in the cloud to communicate with a NetBackup master server that is behind a firewall.

NBWSS uses the WebSocket protocol to create a secure connection to the application's server in the cloud. On that connection, the application can interact with NetBackup by invoking REST APIs and can receive notifications from NetBackup.

NetBackup communicates with the cloud-based application over a web interface that the cloud application makes available. That interface is called a WebSocket endpoint. When a connection exists between NetBackup and the cloud application's endpoint, the application can use NBWSS messages to direct NetBackup to perform data protection services.

Figure 1-1   NBWSS overview
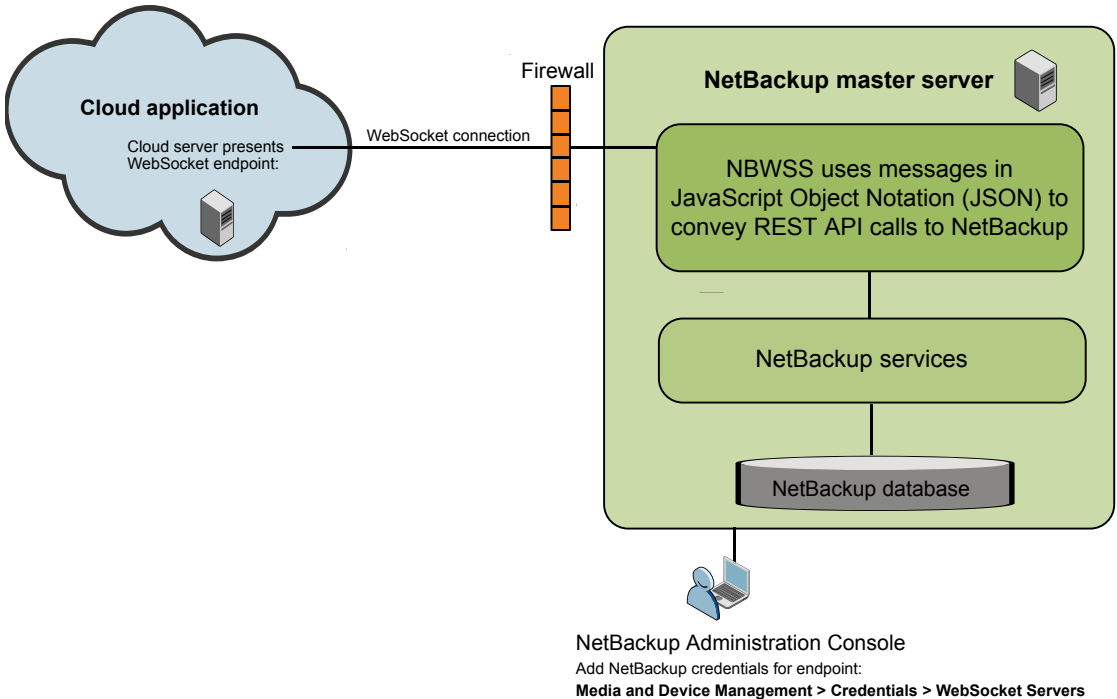
Table 1-1 lists the phases in the NBWSS communication process.

Table 1-1   NBWSS communication process

Phase	Description
Phase 1	With its endpoint credentials, NetBackup sends a connection request to the cloud application. For example: See "NetBackup requests a connection to the endpoint" on page 16.
Phase 2	The cloud application reads the connection request and sends NetBackup a response. For example: See "NetBackup requests a connection to the endpoint" on page 16.
Phase 3	When a connection is established, the cloud application's NBWSS component can call NetBackup APIs to perform data protection services (such as backup or recovery). The cloud application must also interpret each response from NetBackup. For example: See "The cloud application asks to make a REST API call" on page 17. See "Notes on NetBackup connections to cloud-applications" on page 9.
Phase 4	NetBackup sends notifications to the cloud application regarding jobs (start and end) and backup images (create, update, delete). The cloud application interprets and acknowledges the notifications. For example: See "NetBackup notification messages for a backup job" on page 19. See "NBWSS notifications" on page 13.

Task overview for setting up NBWSS communication

Table 1-2 lists the tasks for setting up NetBackup communication with a cloud-based application.

Table 1-2   Setting up NBWSS communication between NetBackup and a cloud-based application

Sequence	Tasks
Task 1	The service provider develops a component in the cloud application that communicates with NetBackup by means of NBWSS messages. For more information, refer to the following topics: See "WebSocket endpoint details and their formatting" on page 30.

	Table 1-2	Setting up NBWSS communication between NetBackup and a cloud-based application *(continued)*

Sequence	Tasks
Task 2	The service provider sends the cloud application's WebSocket endpoint details to the NetBackup administrator.
Task 3	To add the endpoint to NetBackup, the NetBackup administrator saves the endpoint details as access credentials. See "Saving NetBackup credentials for a WebSocket server endpoint" on page 33.
Task 4	The NetBackup administrator can adjust the properties of NBWSS. For example, you can change the time interval at which NetBackup starts a new connection to a cloud application. See "Configuring the properties of the NetBackup WebSocket Service (NBWSS)" on page 41. See "Starting a NetBackup connection to a cloud application" on page 44.

Notes on NetBackup connections to cloud-applications

NBWSS uses the following rules to establish a connection to an endpoint:

- If no active connections exist to an endpoint in a server group, NetBackup attempts to connect to the endpoint that has the highest priority.

- If unable to connect to an endpoint within a server group (server is down), NetBackup attempts to connect to the endpoint that has the next highest priority in that server group.

Note these additional rules and limitations:

- At most one connection can exist per endpoint at a time.

- At most one connection can exist per server group at a time.

- NBWSS does not automatically close an existing connection when a higher priority connection comes online. For example, assume that server group sg1 has two endpoints (ep1 and ep2) with priorities 1 and 2, respectively. If NBWSS is currently connected to ep2 (priority 2) and ep1 (priority 1) comes online, NBWSS does not automatically connect to ep1. The cloud application must close the connection to ep2 before NBWSS attempts to connect to ep1.

- A connection process runs on a timer that responds to endpoint connection changes (such as for connecting to new endpoints or disconnecting from removed endpoints). The default period for this task is 60 seconds. As a result, it may be up to 1 minute before endpoint changes take effect.

You can use the `connectionInfo.period` property to configure this task.

See "Configuring the properties of the NetBackup WebSocket Service (NBWSS)" on page 41.

- When the NetBackup Web Management Console service is restarted, the NetBackup web server takes a few minutes to start. As a result, it takes a few minutes for currently configured endpoints to appear in the NetBackup Administration Console. The endpoints appear under **Media and Device Management > Credentials > WebSocket Servers**.

- An established connection does not have a time limit; the connection can exist indefinitely. In some cases the connection may have to be re-established, such as when the token that NetBackup sends to the cloud application has expired. In that case, the NetBackup credentials for the endpoint must be re-added with a new, valid token. The connection is re-established the next time the connection process runs (determined by the `connectionInfo.period` property).

NBWSS message formats

To communicate with WebSocket endpoints, the NetBackup WebSocket Service (NBWSS) uses its own message format with JavaScript Object Notation (JSON). The JSON format allows NBWSS and the applications on the endpoints to keep track of messages by ID and determine their type and subtype.

The messages operate as request and response: each request has an associated response.

The following is an example of an NBWSS connection request:

```
{
     "version": "1.0",
     "id": "0CEAB6C2-0BBF-4F60-974D-C1F3EF39B872",
     "type": "CONNECT",
     "subType": "REQUEST",
     "timeStamp": 1444944181,
     "payload": {
          "token": "qwerrtrtrtrt2234344==="
     }
}
```

An example of an application's response:

```
{
     "version": "1.0",
     "id": "0CEAB6C2-0BBF-4F60-974D-C1F3EF39B872",
     "type": "CONNECT",
```

```
        "subType": "RESPONSE",
        "timeStamp": 1444944191,
        "payload": {
              "valid": true
        }
}
```

Note the following:

- The message begins with a left curly bracket ({) and ends with a right curly bracket (}).

- The response should have the same value for `"id"` as the request.

- The entries consist of `key:value` pairs that are comma-separated.

- The message includes a `payload`. For messages of type `CONNECT` or `COMMAND`, the payload contains an *object* within curly brackets { }. For messages of type `NOTIFICATION`, the payload contains an *array* within square brackets [ ].

- For background on JSON formatting, see the Network Working Group memo on JavaScript Object Notation:
  http://www.ietf.org/rfc/rfc4627.txt?number=4627

Table 1-3 describes the fields in the NBWSS messages.

Table 1-3   NBWSS message fields

Key	Description
version:	The version of the message. In this release, the available version is `1.0`.
id:	A unique identifier for the message.
	When NBWSS sends a request message, it generates a UUID and places it in this field. When the application at the endpoint responds with a response message, NBWSS expects the response to contain the same ID as the request message. The ID allows NBWSS to map the request message to the response message.
	When NBWSS receives a request message, its response message contains the same ID as the request message. The ID allows the endpoint application to map the request to the response if necessary.

Table 1-3 NBWSS message fields *(continued)*

Key	Description
type:	The message type. The available types are: ■ CONNECT To request a connection to an endpoint. ■ COMMAND To request the execution of a REST API call. ■ NOTIFICATION To report on the status of NetBackup events, such as the status of a backup job.
subType:	The message subtype. The available subtypes are REQUEST or RESPONSE.
timeStamp:	A numeric representation of the UNIX Epoch time (in seconds) when the message was sent.
payload:	The body of the message. The body varies with the type and subtype of the message. The following topics include further details and examples: See "API calls over NBWSS" on page 12. See "NBWSS notifications" on page 13. See "Examples of NBWSS messages" on page 16.

API calls over NBWSS

The NetBackup WebSocket Service (NBWSS) allows a cloud-based application to make REST API calls to NetBackup over a secure connection. The cloud application sends messages to NBWSS in JavaScript Object Notation (JSON). The JSON messages contain the REST API call that the cloud application wants to execute. NBWSS then makes the API call on the cloud application's behalf and sends back a response to the application.

The following is an example request to make a NetBackup REST API call:

```
{
    "version": "1.0",
    "id": "9CD2B69F-0BBF-3F60-974D-C1F2EF37B872",
    "type": "COMMAND",
    "subType": "REQUEST",
    "timeStamp": 1444806222,
```

```
            "payload": {
                "uri": "/netbackup/config/servers/vmservers/vCenter1.domain
                        .com",
                "method": "GET",
                "headers": {
                        "Content-Type": "application/json"
                }
            }
        }
}
```

Note the following:

- To make an API call, the `"type"` field must be `"COMMAND"` and the `"subType"` field must be `"REQUEST"`.

- The `"payload"` field depends on the type of API to be called.

  - In this example, the `"uri"` field contains the URI of the REST API call. NBWSS makes sure that the host name and port are properly included in the full REST request.

  - The `"method"` field indicates the type of API call to be made. In this example, it is `"GET"` (a request to get information about vCenter1).

  - The `"headers"` field contains any HTTP headers to include with the API call. In this example, `"Content-Type"` is set to `"application/json"`, to indicate that the request is sent in JSON format.

NBWSS notifications

When NetBackup is connected to an NBWSS endpoint, the endpoint receives notifications from NetBackup in the form of a `NOTIFICATION REQUEST` message. When the endpoint receives the notification, the endpoint should respond with a `NOTFICATION RESPONSE` message.

Table 1-4 describes the types of notifications that NetBackup sends.

Table 1-4   NetBackup notification types

Notification types	Description
NetBackup job notifications	When a job starts, NetBackup issues a notification of the job's current state: `"QUEUED"`, `"ACTIVE"`, or `"DONE"`. Note that NetBackup polls for the job's state at regular intervals.
	When a job completes, NetBackup issues a notification that the job's state is `"DONE"`. NetBackup issues this notification whether the job succeeded or failed.

Table 1-4 NetBackup notification types *(continued)*

Notification types	Description
NetBackup backup image notifications	When NetBackup creates a backup image, it issues a notification that the image state is "CREATE" or "UPDATE".
	When a backup image is updated, NetBackup issues a notification that the image state is "UPDATE".
	When a backup image is deleted, NetBackup issues a notification that the image state is "DELETE".
	When an image copy expires, if all remaining local copies are replica copies that cannot be restored, NetBackup issues the notification "NO_LOCAL_COPY_AVAILABLE".

Notification message format

A. Notification Request

NetBackup sends notifications to an endpoint in the form of a NOTIFICATION REQUEST message. This message may have one or more notifications within its payload.

The following is an example of a notification request:

```
{
        "version": "1.0",
        "id": "EDD85CD7-8553-47E4-8A19-01C65092F220",
        "type": "NOTIFICATION",
        "subType": "REQUEST",
        "timeStamp": 1459811679,
        "payload": [
            {
                "notificationType": "INFO",
                "object": "JOB",
                "data": [
                    {
                        "scheduleType": "ST_FULL",
                        "clientName": "mserver2.acme.com",
                        "status": 0,
                        "startTime": 1459829674,
                        "state": "ACTIVE",
                        "policyName": "vmware2",
                        "parentJobId": 144,
                        "jobId": 144,
                        "policyType": "VMWARE",
                        "jobType": "BACKUP"
```

```
                    }
                ]
            }
        ]
}
```

In request messages, the "payload" value type is an array. Each element of the array holds a different notification object type ("JOB" or "IMAGE"). The element has all notifications that are related to that object type. This array allows NetBackup to batch together notifications of a similar type.

For example, for job start and job done, the payload has one element: a notification object of type "JOB". Within the data section of the "JOB" notification object, there are two elements, one for each notification. For an example of batched notifications in one message, see "Multiple notifications in one message" in the following topic:

See "Other NetBackup notification messages" on page 24.

Each notification object has the following fields:

- notificationType:
  A string that displays the type of notification. In this release, the only type is "INFO".

- object:
  A string that displays the notification's object. In this release, the only objects are "JOB" and "IMAGE".

- data:
  An array that contains the information for each object type. Each data array element is a separate notification. The fields in the data array are specific to each type of notification.
  See "Examples of NBWSS messages" on page 16.

B. Notification Response

For each notification request, a NOTIFICATION RESPONSE message is expected. The "id" field of this response should be the same as the "id" of the request and the "payload" field should be an empty array.

For example:

```
{
    "version": "1.0",
    "id": "EDD85CD7-8553-47E4-8A19-01C65092F220",
    "type": "NOTIFICATION",
    "subType": "RESPONSE",
    "timeStamp": 1445036999,
```

```
        "payload": []
}
```

When NetBackup receives the response, the notifications that were sent within the request are considered acknowledged and new notifications can then be sent as they occur. If a notification request is not acknowledged within the configured time period, the notification is resent. No new notifications are sent to that endpoint until the notification is acknowledged.

The time period can be configured in the `nbwss.properties` file by means of the `notification.scheduledRate` option. The default is 5 seconds. The following topic contains more information on the options in the `nbwss.properties` file:

See "Configuring the properties of the NetBackup WebSocket Service (NBWSS)" on page 41.

Guaranteed delivery

To avoid delivery problems, NetBackup guarantees delivery of notifications in the following cases: the connection between NetBackup and the endpoint drops, the endpoint server goes offline, or a problem occurs with NetBackup Web Services. If an endpoint server is offline, the notifications go to the next endpoint server in the server group.

See "Notes on NetBackup connections to cloud-applications" on page 9.

Examples of NBWSS messages

The following are examples of NBWSS messages and notifications, with explanatory notes.

NetBackup requests a connection to the endpoint

A. NetBackup initiates the connection request

```
{
        "version": "1.0",
        "id": "0CEAB6C2-0BBF-4F60-974D-C1F3EF39B872",
        "type": "CONNECT",
        "subType": "REQUEST",
        "timeStamp": 1444944181,
        "payload": {
                "token": "qwerrtrtrtrt2234344==="
        }
}
```

Notes: In this message, the `"type"` field is `"CONNECT"` and the `"subType"` is `"REQUEST"`. The `"token"` key contains the application validation token that was added when the endpoint was configured in NetBackup. The cloud-based application validates this token and sends a `CONNECT RESPONSE` message with the results of the validation (see the following example).

B. The endpoint responds to NetBackup's request

The `"subType"` is `"RESPONSE"`.

```
{
      "version": "1.0",
      "id": "0CEAB6C2-0BBF-4F60-974D-C1F3EF39B872",
      "type": "CONNECT",
      "subType": "RESPONSE",
      "timeStamp": 1444944191,
      "payload": {
            "valid": true
      }
}
```

Notes: If the token is validated, the application responds with the `"valid"` field set to `true`. NetBackup then considers the connection to be established and operations can proceed. If the token is not valid, the application should respond with `"valid"` set to `false`, which causes NetBackup to close the connection.

Note: The response should always have the same `"id"` as the request.

The cloud application asks to make a REST API call

A. The cloud application asks to add information to NetBackup about a vCenter server (POST)

```
{
      "version": "1.0",
      "id": "99B9BD8C-9E3E-406A-A7EE-33B88531C7D9",
      "type": "COMMAND",
      "subType": "REQUEST",
      "timeStamp": 1444856264,
      "payload": {
            "uri": "/netbackup/config/servers/vmservers",
            "method": "POST",
            "headers": {
```

```
                       "Content-Type": "application/json"
                       "Authorization": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1N"
               },
               "parameters": "{\"serverName\": \"vcenterServer1\",
               \"proxyServerName\": \"\",\"vmType\": \"VMWARE_VIRTUAL_
               CENTER_SERVER\",\"userId\": \"administrator\",
               \"password\": \"password@123\",\"port\": 0 }"
           }
     }
```

Notes: The request and its response should always have the same value for `"id"`.

The `"type"` field is `"COMMAND"` and the `"subType"` field is `"REQUEST"`. The `"payload"` `"method"` is `"POST"`, which adds the `vcenterServer1` information into NetBackup.

For `"subType"` `"REQUEST"`, the `"headers"`: must contain the following:

- `"Content-Type": "application/json"` is the form of the request.

- `"Authorization"` is the JSON web token (JWT) that was received in a previous response.

The `"parameters"` field is a JSON-escaped string: the double quotes around each value (such as `"serverName"`) are escaped with a backslash (\).

B. The cloud application asks to read information about a vCenter server (GET)

```
{
     "version": "1.0",
     "id": "9CD2B89F-0BBF-4F60-974D-C1F3EF39B872",
     "type": "COMMAND",
     "subType": "REQUEST",
     "timeStamp": 1444806222,
     "payload": {
           "uri": "/netbackup/config/servers/vmservers/vCenter2
           .domain.com",
           "method": "GET",
           "headers": {
                   "Content-Type": "application/json"
                   "Authorization": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1N"
           }
     }
}
```

Notes: The `"type"` field is `"COMMAND"` and the `"subType"` field is `"REQUEST"`. The `"payload"` `"method"` is `"GET"`, which reads information about `vCenter2.domain.com` that is stored in NetBackup.

C. NetBackup responds to the endpoint request

```
{
      "version": "1.0",
      "id": "9CD2B89F-0BBF-4F60-974D-C1F3EF39B872",
      "type": "COMMAND",
      "subType": "RESPONSE",
      "timeStamp": 1444806444,
      "payload": {
            "headers": {
                  "date": "Thu, 14 Jan 2016 20:58:11 GMT",
                  "cache-control": "private",
                  "server": "Apache-Coyote/1.1",
                  "content-type": "application/json",
                  "transfer-encoding": "chunked",
                  "expires": "Wed, 31 Dec 1969 16:00:00 PST"
            },
            "responseCode": 200,
            "body": "{\"vmServer\":{\"serverName\":\"vCenter2.domain
            .com\",\"vmType\":\"VMWARE_VIRTUAL_CENTER_SERVER\",
            \"userId\":\"root\",\"password\":\"\",\"port\":0},
            \"links\":[{\"rel\":\"self\",\"href\":\"https://xuanbl5vm9:
            8443/config/servers/vmservers/vCenter2.domain.com\"}]}"
      }
}
```

Notes:

The `"payload"` contains the HTTP response (`"headers"`, `"response code"`, and `"body"`) that NetBackup received from the API.

NetBackup notification messages for a backup job

Examples **A** through **G** are the notifications that NetBackup sent to an endpoint for a backup from a VMware Intelligent Policy.

A. Start of a parent backup job (discovery)

```
{
      "version": "1.0",
      "id": "EDD85CD7-8555-47E4-8A19-01C35093F220",
```

```
        "type": "NOTIFICATION",
        "subType": "REQUEST",
        "timeStamp": 1459811679,
        "payload": [
                {
                        "notificationType": "INFO",
                        "object": "JOB",
                        "data": [
                                {
                                        "scheduleType": "ST_FULL",
                                        "clientName": "masterserver2.domain.com",
                                        "status": 0,
                                        "startTime": 1459829674,
                                        "state": "ACTIVE",
                                        "policyName": "vmware2",
                                        "parentJobId": 144,
                                        "jobId": 144,
                                        "policyType": "VMWARE",
                                        "jobType": "BACKUP"
                                }
                        ]
                }
        ]
}
```

B. Start of the child snapshot job

```
{
        "version": "1.0",
        "id": "7C0FD14E-089E-46C8-AA2B-344D69AA0C67",
        "type": "NOTIFICATION",
        "subType": "REQUEST",
        "timeStamp": 1459811689,
        "payload": [
                {
                        "notificationType": "INFO",
                        "object": "JOB",
                        "data": [
                                {
                                        "scheduleType": "ST_FULL",
                                        "clientName": "DummyTestVM",
                                        "status": 0,
                                        "startTime": 1459829686,
                                        "state": "ACTIVE",
```

```
                                    "policyName": "vmware2",
                                    "parentJobId": 144,
                                    "jobId": 145,
                                    "policyType": "VMWARE",
                                    "jobType": "BACKUP"
                        }
                ]
        }
    ]
}
```

C. Start of child backup job (actual backup)

```
{
    "version": "1.0",
    "id": "EF507ECE-4B1C-4D87-AAB0-032ADBC915FC",
    "type": "NOTIFICATION",
    "subType": "REQUEST",
    "timeStamp": 1459811704,
    "payload": [
        {
                "notificationType": "INFO",
                "object": "JOB",
                "data": [
                        {
                                "scheduleType": "ST_FULL",
                                "clientName": "DummyTestVM",
                                "status": 0,
                                "startTime": 1459829698,
                                "state": "ACTIVE",
                                "policyName": "vmware2",
                                "parentJobId": 145,
                                "jobId": 146,
                                "policyType": "VMWARE",
                                "jobType": "BACKUP"
                        }
                ]
        }
    ]
}
```

D. Image creation

```
{
     "version": "1.0",
     "id": "608FE0C1-B03C-421D-8876-E3730A7855AF",
     "type": "NOTIFICATION",
     "subType": "REQUEST",
     "timeStamp": 1459811724,
     "payload": [
          {
               "notificationType": "INFO",
               "object": "IMAGE",
               "data": [
                    {
                         "clientType": "VMWARE",
                         "clientName": "DummyTestVM",
                         "backupTime": 1459811698,
                         "createdTime": 1459829720,
                         "operationId": "CREATE",
                         "backupId": "DummyTestVM_1459811698"
                    },
                    {
                         "clientType": "VMWARE",
                         "clientName": "DummyTestVM",
                         "backupTime": 1459811686,
                         "createdTime": 1459829721,
                         "operationId": "UPDATE",
                         "backupId": "DummyTestVM_1459811686"
                    }
               ]
          }
     ]
}
```

E. Backup job complete (actual backup job)

```
{
     "version": "1.0",
     "id": "608FE0C1-B03C-421D-8876-E3730A7855AF",
     "type": "NOTIFICATION",
     "subType": "REQUEST",
     "timeStamp": 1459811724,
     "payload": [
          {
```

```
                          "notificationType": "INFO",
                          "object": "JOB",
                          "data": [
                                {
                                        "scheduleType": "ST_FULL",
                                        "clientName": "DummyTestVM",
                                        "status": 0,
                                        "startTime": 1459829698,
                                        "state": "DONE",
                                        "policyName": "vmware2",
                                        "parentJobId": 145,
                                        "jobId": 146,
                                        "policyType": "VMWARE",
                                        "jobType": "BACKUP"
                                }
                          ]
                  }
          ]
  }
```

F. Snapshot job complete

```
{
      "version": "1.0",
      "id": "F97BAE8F-D1E3-4242-A5EC-FB1C9B8F46E3",
      "type": "NOTIFICATION",
      "subType": "REQUEST",
      "timeStamp": 1459811734,
      "payload": [
            {
                    "notificationType": "INFO",
                    "object": "JOB",
                    "data": [
                          {
                                  "scheduleType": "ST_FULL",
                                  "clientName": "DummyTestVM",
                                  "status": 0,
                                  "startTime": 1459829686,
                                  "state": "DONE",
                                  "policyName": "vmware2",
                                  "parentJobId": 144,
                                  "jobId": 145,
                                  "policyType": "VMWARE",
                                  "jobType": "BACKUP"
```

```
                    }
                ]
            }
        ]
}
```

G. Parent backup job complete

```
{
        "version": "1.0",
        "id": "F97BAE8F-D1E3-4242-A5EC-FB1C9B8F46E3",
        "type": "NOTIFICATION",
        "subType": "REQUEST",
        "timeStamp": 1459811734,
        "payload": [
            {
                "notificationType": "INFO",
                "object": "JOB",
                "data": [
                    {
                        "scheduleType": "ST_FULL",
                        "clientName": "masterserver2.domain.com",
                        "status": 0,
                        "startTime": 1459829674,
                        "state": "DONE",
                        "policyName": "vmware2",
                        "parentJobId": 144,
                        "jobId": 144,
                        "policyType": "VMWARE",
                        "jobType": "BACKUP"
                    }
                ]
            }
        ]
}
```

Other NetBackup notification messages

The following messages are the notifications that NetBackup sent to an endpoint for a restore job and for image deletion. The third message is an example of multiple notifications in one message.

Restore job done

```
{
      "version": "1.0",
      "id": "8E909940-AD50-4543-8AEA-B52003818925",
      "type": "NOTIFICATION",
      "subType": "REQUEST",
      "timeStamp": 1459812309,
      "payload": [
            {
                  "notificationType": "INFO",
                  "object": "JOB",
                  "data": [
                        {
                              "scheduleType": "ST_FULL",
                              "clientName": "masterserver2.domain.com",
                              "status": 0,
                              "startTime": 1459830185,
                              "state": "DONE",
                              "policyName": "",
                              "parentJobId": 147,
                              "jobId": 147,
                              "policyType": "STANDARD",
                              "jobType": "RESTORE"
                        }
                  ]
            }
      ]
}
```

Image deletion

```
{
      "version": "1.0",
      "id": "15AAF7BA-C082-4996-A55D-7C4745D4D1E9",
      "type": "NOTIFICATION",
      "subType": "REQUEST",
      "timeStamp": 1459814495,
      "payload": [
            {
                  "notificationType": "INFO",
                  "object": "IMAGE",
                  "data": [
                        {
```

```
                                    "clientType": "VMWARE",
                                    "clientName": "localhost",
                                    "backupTime": 1458601200,
                                    "createdTime": 1459832492,
                                    "operationId": "DELETE",
                                    "backupId": "localhost_1458601200"
                            }
                        ]
                }
        ]
}
```

Note: If the NetBackup master server uses Auto Image Replication (AIR), the following notification may be issued regarding image deletion:

```
{
  "version": "1.0",
  "id": "E38DD102-98BC-4590-8E09-85B0A0EA31CE",
  "type": "NOTIFICATION",
  "subType": "REQUEST",
  "timeStamp": 1471471464,
  "payload": [
    {
      "notificationType": "INFO",
      "object": "IMAGE",
      "data": [
        {
          "clientType": "STANDARD",
          "clientName": "localhost",
          "backupTime": 1471469619,
          "createdTime": 1471485862,
          "operationId": "UPDATE",
          "backupId": "localhost_1471469619"
        },
        {
          "clientType": "STANDARD",
          "clientName": "localhost",
          "backupTime": 1471469619,
          "createdTime": 1471485862,
          "operationId": "NO_LOCAL_COPY_AVAILABLE",
          "backupId": "localhost_1471469619"
        }
      ]
    }
```

```
        ]
}
```

Multiple notifications in one message

```
{
      "version": "1.0",
      "id": "608FE0C1-B03C-421D-8876-E3730A7855AF",
      "type": "NOTIFICATION",
      "subType": "REQUEST",
      "timeStamp": 1459811724,
      "payload": [
            {
                  "notificationType": "INFO",
                  "object": "JOB",
                  "data": [
                        {
                              "scheduleType": "ST_FULL",
                              "clientName": "DummyTestVM",
                              "status": 0,
                              "startTime": 1459829698,
                              "state": "DONE",
                              "policyName": "vmware2",
                              "parentJobId": 145,
                              "jobId": 146,
                              "policyType": "VMWARE",
                              "jobType": "BACKUP"
                        }
                  ]
            },
            {
                  "notificationType": "INFO",
                  "object": "IMAGE",
                  "data": [
                        {
                              "clientType": "VMWARE",
                              "clientName": "DummyTestVM",
                              "backupTime": 1459811698,
                              "createdTime": 1459829720,
                              "operationId": "UPDATE",
                              "backupId": "DummyTestVM_1459811698"
                        },
                        {
                              "clientType": "VMWARE",
```

```
                                "clientName": "DummyTestVM",
                                "backupTime": 1459811686,
                                "createdTime": 1459829721,
                                "operationId": "UPDATE",
                                "backupId": "DummyTestVM_1459811686"
                    }
                ]
            }
        ]
    }
```

The following topic contains additional information on notifications:

See "NBWSS notifications" on page 13.

Configuring WebSocket endpoints for NBWSS

This chapter includes the following topics:

- About NetBackup connections to WebSocket endpoints

- WebSocket endpoint details and their formatting

- Saving NetBackup credentials for a WebSocket server endpoint

- WebSocket Server dialog

- Removing NetBackup credentials for a WebSocket server endpoint

- Configuring the properties of the NetBackup WebSocket Service (NBWSS)

- Starting a NetBackup connection to a cloud application

About NetBackup connections to WebSocket endpoints

To establish a connection to a cloud-based application, NetBackup communicates with a web interface that the cloud application makes available. That interface is called a WebSocket endpoint. For the connection, NetBackup needs certain information about the endpoint. Table 2-1 describes the steps for preparing that information.

Table 2-1 Preparing NetBackup credentials for connection to a cloud application endpoint

Task	Description
Obtain the endpoint details.	Contact the cloud service provider for the endpoint information. The following topic describes the required endpoints details: See "WebSocket endpoint details and their formatting" on page 30.
If necessary, format the endpoint details for NetBackup.	The endpoint information must be available to NetBackup in either of the following ways: ■ In a text file that uses JavaScript Object Notation (JSON). ■ In a URL that the service provider generates. NetBackup uses the URL to request the endpoint information. The following topic describes how to save the details in a JSON-formatted file: See "WebSocket endpoint details and their formatting" on page 30.
Save the endpoint details as NetBackup credentials.	In the NetBackup Administration Console, use the **Media and Device Management > Credentials > WebSocket Servers** option to save NetBackup credentials for the cloud application endpoint. See "Saving NetBackup credentials for a WebSocket server endpoint" on page 33.

WebSocket endpoint details and their formatting

To communicate with a cloud-based application, NetBackup uses the WebSocket protocol to establish a secure connection to the cloud application. NetBackup connects to a cloud application interface that is called a WebSocket endpoint. To connect, NetBackup needs certain details about the endpoint.

Table 2-2 describes the information that is required for a WebSocket endpoint.

Table 2-2   Entries that define a WebSocket endpoint

Endpoint details	Description
token	The cloud application's security token.
	When NetBackup initiates a connection to the cloud application, it sends the token to the application. The application then validates the token. If the application accepts the token, a secure connection is established between NetBackup and the application. If the application does not accept the token, the connection is not established.
priority	The endpoint's priority within its group. A lower number has higher priority.
	The priority allows NetBackup to decide in which order to attempt connections for that server group. Only one connection can be active per server group.
groupId	A unique identifier of the group that the endpoint belongs to.
hostName	The host name or IP address of the cloud server that contains the endpoint.
url	The full URL of the WebSocket endpoint that NetBackup connects to.
	The WebSocket URL begins with wss://
	Note: ws:// is not supported.

IMPORTANT: You may need to work with the cloud service provider to obtain the endpoint details. The endpoint details must be available to NetBackup in either of the following ways:

■ In a file that is formatted in JavaScript Object Notation (a JSON file). If the service provider does not provide the endpoint details in a JSON file, you can format the information in a JSON file yourself.

Note: The endpoint details must include a security token for access to the cloud application. The service provider should be careful to send you the application token in a secure manner.

■ By means of a URL. NetBackup uses the URL to request the endpoint details from the cloud application.

Note: NetBackup does not support an apostrophe (') anywhere in the endpoint details.

WebSocket endpoint details in a JSON file

The following shows the WebSocket endpoint details in JavaScript Object Notation (JSON):

```
{
     "token": "security_token ....",
     "priority": numeric_value,
     "groupId": "group_ID",
     "hostName": "host_name.domain",
     "url": "wss://host_name.domain:port/uri"
}
```

Note the following:

- In this version of NetBackup, each JSON file must specify a single endpoint, not multiple endpoints.

- The file begins with a left curly bracket ({) and ends with a right curly bracket (}).

- The entries consist of `name:value` pairs that are comma-separated.

- Each string is enclosed in double quotes ("") except for the priority value.

- The five `name:value` pairs (`token`, `priority`, `groupID`, `hostName`, `url`) can appear in any order.

- NetBackup does not support an apostrophe (') anywhere in the file.

- Save the JSON-formatted information as a text file in a location that the NetBackup master server can access.

- For background on JSON formatting, see the Network Working Group memo on JavaScript Object Notation:
  http://www.ietf.org/rfc/rfc4627.txt?number=4627

The following is an example of a JSON-formatted file that defines a WebSocket endpoint:

```
{
     "token": "MIID4TCCAsmgAwIBAgIEBZCDRzANBgkqhkiG9w0BAQsFADBxMQs
DVQQGEwJVUzELMAkGA1UECBMCQ0ExFjAUBgNVBAcTDU1vdW50YWluIFZpZXcx
vzu0n2rWon48ncp6jMjOFiWqMRXnV8Q0vOEpAzUV7Qml92EMV6z0PinAgMBAA
GjgYAwfjBdBgNVHREEVjBUgiJ2b21yaGVsNnU1LXZtMDQuZW5nYmEuc3ltYW
G7IsZ2fTDWKLGxbAG5NNKwEfD11LFhKGwaHkOXYkVi+HVnFEFKK0gxVWg==",
     "priority": 1,
     "groupId": "GROUPID1",
     "hostName": "vrhel6u5-vm4.acme.com",
```

```
        "url": "wss://vrhel6u5-vm4.acme.com:14146/cfs/nbufacade"
}
```

Notes on the JSON file example:

- This example begins with the token. The token is a string that the cloud application uses to authenticate NetBackup when NetBackup requests a connection.

  Caution: When you obtain the endpoint information from the service provider, make sure that the token is provided in a secure manner.

- The next entry in the file is the `priority`, followed by the `groupId`, `hostName`, and the cloud server's `url`.

When you have the JSON formatted file, use the **FILE** option on the NetBackup **WebSocket Server** dialog to specify that file. NetBackup extracts the endpoint details from the file. Use the following procedure:

See "Saving NetBackup credentials for a WebSocket server endpoint" on page 33.

WebSocket endpoint details obtained over the web

The cloud application can generate a URL that NetBackup can use to request the WebSocket endpoint details. Use the following procedure to enter the URL in the NetBackup **WebSocket Server** dialog:

See "Saving NetBackup credentials for a WebSocket server endpoint" on page 33.

# Saving NetBackup credentials for a WebSocket server endpoint

Use the following procedure to select the JSON file or URL so that NetBackup can save the endpoint details as credentials.

**To save NetBackup credentials for a WebSocket server endpoint**

1 In the NetBackup Administration Console, click **Media and Device Management > Credentials > WebSocket Servers**.

2 Click **Actions > New > New WebSocket Server**.
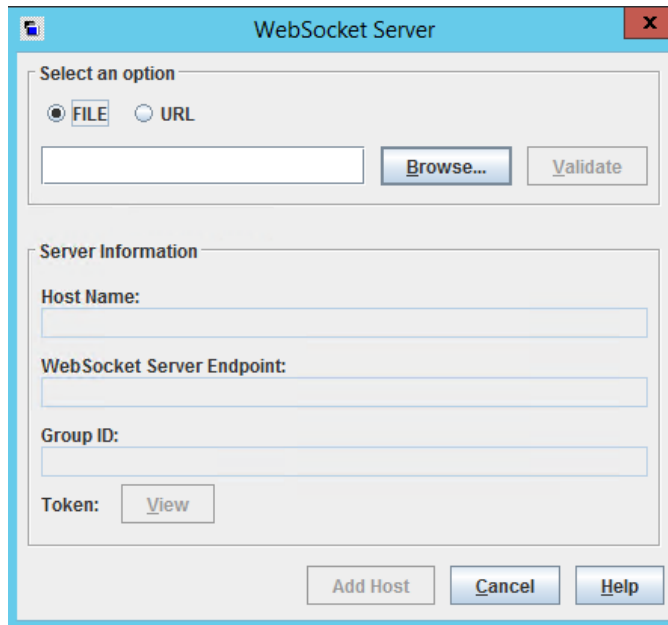
3 In the **WebSocket Server** dialog, select the source of the endpoint details:

   ■ For a JSON-formatted file, click **FILE**, then click **Browse**.
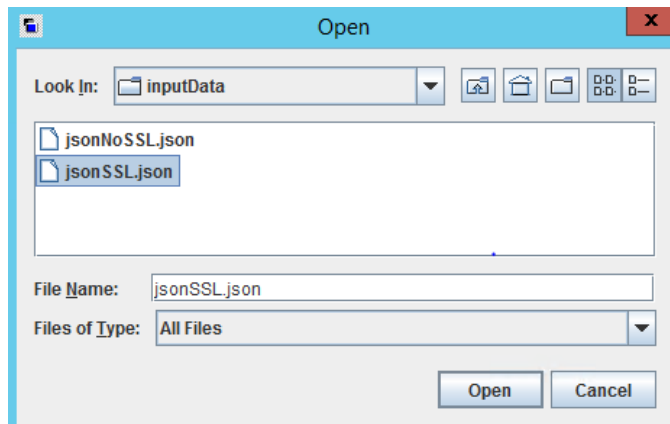
You can enter the file's full path in the **File Name** field, or use the **Look In** pull-down or the search icons. From left to right, the icons can move up one level, go to the desktop, create a new folder, or change the list view.

Next, click on the JSON file and then click **Open**. NetBackup extracts the endpoint details and displays them under **Server Information**:
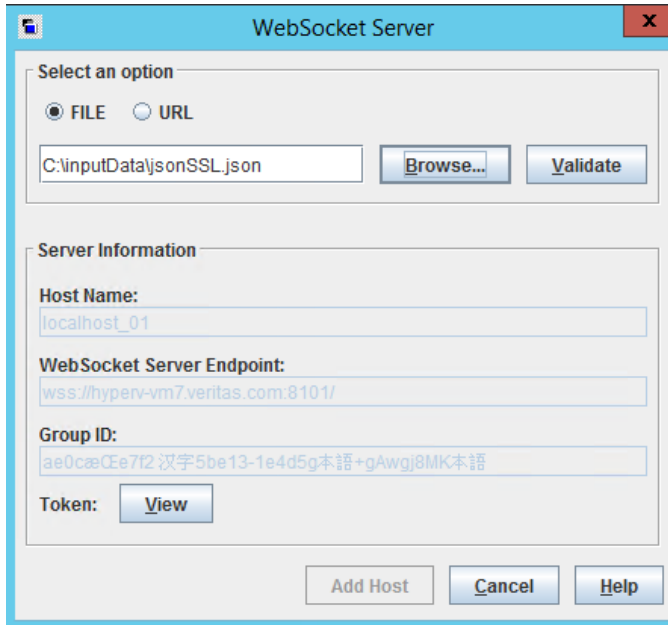
- For a URL, click **URL** and enter the URL that contains the endpoint information. NetBackup extracts the endpoint information from the URL. (The cloud application provides the URL.)
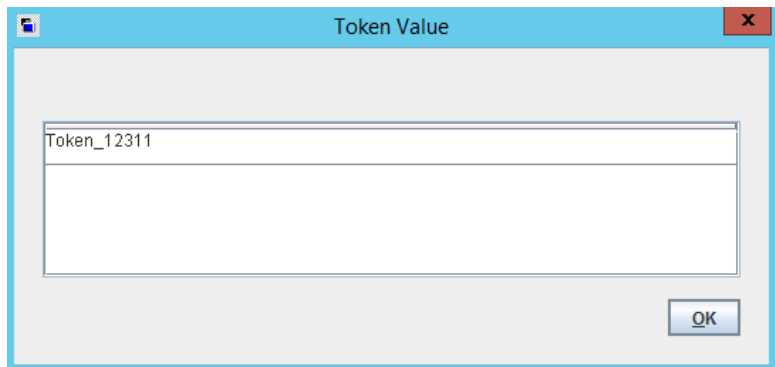
4   Click **Validate**.

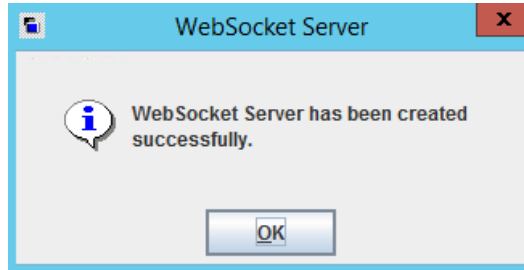NetBackup presents the endpoint server's SSL certificate. For example:

If you used the **URL** option, NetBackup extracts the endpoint details and displays them under **Server Information**.

**5**   To see the cloud application's security token, click **Token: View**.

6   To save this endpoint information as NetBackup credentials, click **Add Host**.
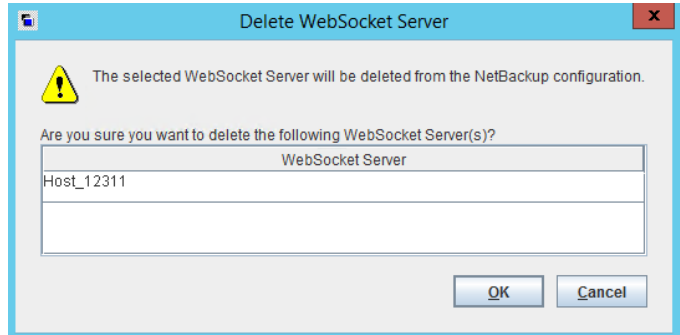
The following appears:

The endpoint's Host Name, URL, Group ID, Priority, State (ACTIVATED or DEACTIVATED), and Connection State (CONNECTED or DISCONNECTED) appear in the right pane under **WebSocket Server(s)**. The endpoint's token is stored in a secure location and is not displayed.
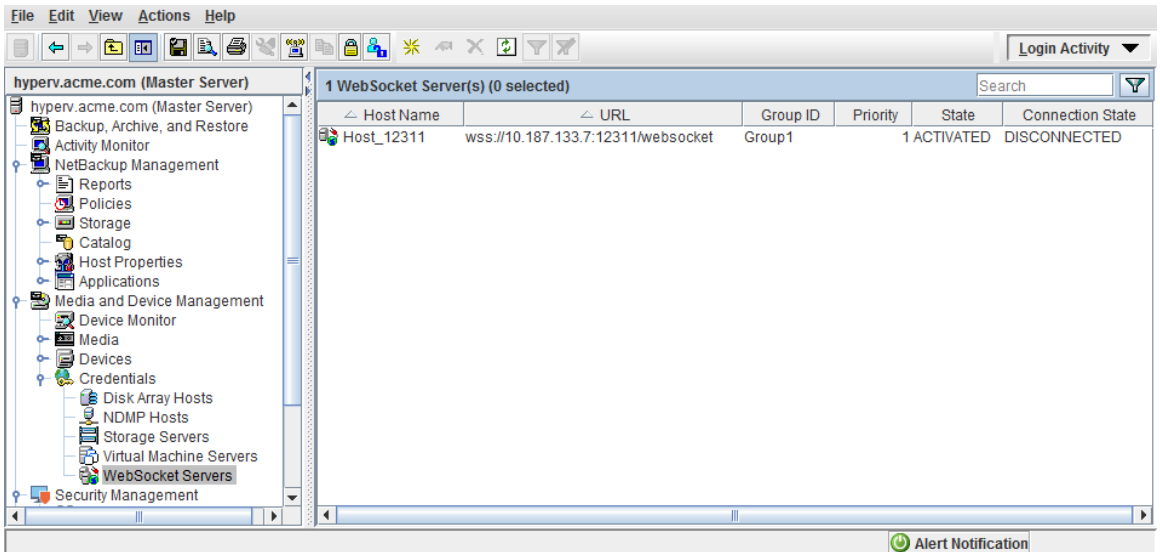
7   To delete or deactivate the endpoint credentials, right-click on the credentials entry in the right pane. The following options are available:

Delete   Removes the endpoint credentials.

	Delete WebSocket Server	☒
⚠	The selected WebSocket Server will be deleted from the NetBackup configuration.	

Are you sure you want to delete the following WebSocket Server(s)?

WebSocket Server
Host_12311

[OK]   [Cancel]

Activate   Activates the endpoint credentials. When the credentials are activated, NBWSS can connect to the endpoint.

Deactivate   Deactivates the endpoint credentials. When the credentials are deactivated, NBWSS does not connect to the endpoint.

WebSocket Server dialog

Use this dialog to save NetBackup credentials for a secure connection to an application server in the cloud. NetBackup uses the server credentials to connect to the cloud server's WebSocket endpoint.

To use this dialog to save the credentials, the endpoint information must be available in either of the following ways:

- In a file that is formatted in JavaScript Object Notation (a JSON file).

- In a URL that the cloud application generated.

The following topic contains a procedure for using this dialog:

Table 2-3   Fields in the **WebSocket Server** dialog

Field	Description
Select an option	Select one of the following to specify the endpoint information: ■ **FILE**: Use this option to locate a JSON-formatted file that contains the endpoint information. **Note:** NetBackup extracts the endpoint information from the file and displays that information in this dialog. ■ **URL**: Use this option to enter the URL that contains the endpoint information. **Note:** NetBackup extracts the endpoint information from the URL and displays that information in this dialog.
Browse	Click **Browse** to locate the JSON-formatted file that contains the endpoint information. Use the **Look In** pull-down or the search icons. From left to right, the icons can move up one level, go to the desktop, create a new folder, or change the list view. As an alternative, you can enter the file's full path in the **File Name** field.
Validate	**REQUIRED**: After you have selected the endpoint information (**FILE** or **URL**), click **Validate** to view the SSL certificate of the endpoint. **Note:** If you entered a URL for the endpoint information, click **Validate** to extract the information and display it under **Server Information**.
Server Information	The following fields show the endpoint information that NetBackup extracted from the JSON file or the URL.
Host Name:	The fully qualified host name or IP address of the cloud server that contains the endpoint. This host name or IP address must be unique: it must not be the host name or IP address for an endpoint that has already been added.
WebSocket Server Endpoint:	The full URL, port, and any additional identifier of the WebSocket endpoint. Example endpoint: `wss://cloudhost7.nebula.com:8080/netbackup/face1` **Note:** `ws://` is not supported.
Group ID:	The server group that the endpoint belongs to.

Table 2-3   Fields in the **WebSocket Server** dialog *(continued)*

Field	Description
Token: View	Click **View** to display the security token that the cloud application uses to validate the identity of NetBackup.
Add Host	If the extracted endpoint information is correct, click **Add Host** to save this information as NetBackup endpoint credentials. At a configurable interval, a scheduled task checks the NetBackup database for updates to endpoints and acts accordingly. It can take up to the configured time (default is 5 minutes) to connect after you add an endpoint. See "Configuring the properties of the NetBackup WebSocket Service (NBWSS)" on page 41.

The following topic describes the endpoint information and its formatting in more detail:

See "WebSocket endpoint details and their formatting" on page 30.

Removing NetBackup credentials for a WebSocket server endpoint

Use the following procedure to remove the endpoint credentials for a cloud application.

To remove NetBackup credentials for a WebSocket endpoint

1   In the NetBackup Administration Console, click **Media and Device Management > Credentials > WebSocket Servers**.

2   In the right pane, right-click on the endpoint credentials, select **Delete**, and click **OK** to confirm the deletion.
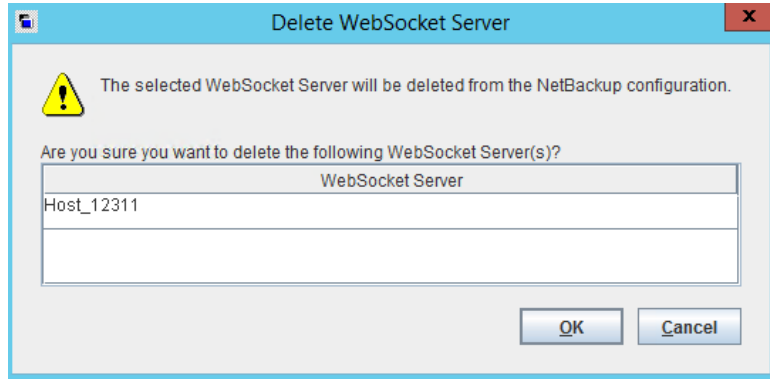
Configuring the properties of the NetBackup WebSocket Service (NBWSS)

Table 2-4 describes the configurable NBWSS properties and their defaults. The properties are in the `nbwss.properties` text file on the NetBackup master server. (For the location of this file, see the procedure in this topic.)

Each property appears on a single line in the following form:

```
key=value
```

An example `.properties` file is included after the table. To change the properties, use the procedure at the end of this topic.

Table 2-4 Configurable properties of NBWSS

Keys	Description
`exception.ignoreDecoder`	Boolean
	Determines whether or not NBWSS ignores a decoder exception. A decoder exception usually occurs when NBWSS is unable to understand a message it receives.
	The default is false: NBWSS closes the connection when a decoder exception occurs.
`notification.sendTimeout=`	Integer
	Determines how long (in milliseconds) NBWSS waits to communicate with an endpoint (to send or receive a notification). By default, a notification task waits for 2 seconds before the task terminates. The notification task tries again after the time limit that the `notification.scheduledRate` property sets.
	The default is 2000 milliseconds (2 seconds). In most cases, Veritas recommends the default.
`notification.scheduledRate=`	Integer
	Determines how often (in seconds) NetBackup queries for new notifications. This value also determines how long NetBackup waits to receive an acknowledgement for a notification before it resends the notification.
	The default is 5 seconds.
`keepAlive.scheduledRate=`	Integer
	Determines how often (in seconds) NBWSS sends a ping to each endpoint as part of its keep alive functionality. If NetBackup receives a pong in response to each ping, the endpoint connection is still valid.
	The default is 30 seconds.
`keepAlive.maxPongMissAllowed=`	Integer
	Determines how many pongs (responses to pings) can be missed for an endpoint connection. When NBWSS sends a ping to an endpoint and a pong is not received, that is considered a missed pong. When the maximum is reached, NBWSS closes the connection to the endpoint.
	The default is 10 missed pongs.

Table 2-4   Configurable properties of NBWSS *(continued)*

Keys	Description
`connectionInfo.period=`	Integer
	Determines the number of seconds between each NBWSS connection update. Each update determines the endpoints that are currently configured in NetBackup, and connects to new endpoints or disconnects from the endpoints that no longer exist.
	Note: After you add an endpoint, it can take up to the configured time to connect to that endpoint.
	The default is 60 seconds.
`scheduledExecutor.threadPoolSize=`	Integer
	Determines how many threads NetBackup uses to maintain the endpoint connection and to handle notifications.
	The default is 1 thread. It may be helpful to increase this value if the number of scheduled tasks increases.
The `hibernate` properties	These properties are for the use of Veritas Support.

Here is the `nbwss.properties` file with its default settings (see the following procedure for the location of this file):

```
#Properties file for NetBackup WebSocket Service
exception.ignoreDecoder=false
notification.sendTimeout=2000
notification.scheduledRate=5
keepAlive.scheduledRate=30
keepAlive.maxPongMissAllowed=10
connectionInfo.period=60
scheduledExecutor.threadPoolSize=1

#Hibernate properties
hibernate.format_sql=true
hibernate.show_sql=false
hibernate.hbm2ddl.auto=update
hibernate.dialect=org.hibernate.dialect.SybaseDialect
```

To configure the properties of the NetBackup WebSocket Service (NBWSS)

1 Use a text editor to open the `nbwss.properties` file.

The file is in the following location on the NetBackup master server:

On Windows:

```
install_path\NetBackup\wmc\webserver\webapps_api\
nbwss\WEB-INF\classes\nbwss.properties
```

On Linux:

```
/usr/openv/wmc/webserver/webapps_api/nbwss/WEB-INF/classes/
nbwss.properties
```

2 Edit the value of the property that you want to change, and save the file.

Table 2-4 describes the NBWSS properties and their defaults.

3 For the changes to take effect, it may be necessary to restart the **NetBackup Web Management Console** service on the NetBackup master server.

Starting a NetBackup connection to a cloud application

To talk to a cloud application, NetBackup uses a web interface that the cloud server makes available. That interface is called a WebSocket endpoint.

A NetBackup process automatically requests a connection to the WebSocket endpoint according to a preset schedule. By default, the connection process runs every 60 seconds. That process is controlled by the `connectionInfo.period=` property in the `nbwss.properties` file on the NetBackup master server. Whenever the connection process runs, it updates (adds or deletes) NetBackup connections to endpoints. For example, if a new endpoint has been added, the process checks if NetBackup is already connected to another endpoint in the same server group. If NetBackup is not connected to another endpoint in the same server group, the process initiates a connection to the new endpoint.

To control the time interval at which NetBackup starts a connection to a cloud application

◆ On the NetBackup master server, edit the `connectionInfo.period=` property in the `nbwss.properties` file.

For the location of this file and further details:

See "Configuring the properties of the NetBackup WebSocket Service (NBWSS)" on page 41.

See "Notes on NetBackup connections to cloud-applications" on page 9.

Note: To start a connection, NetBackup must have the proper credentials to access the cloud server's endpoint:

See "WebSocket endpoint details and their formatting" on page 30.

See "Saving NetBackup credentials for a WebSocket server endpoint" on page 33.

Troubleshooting NBWSS

This chapter includes the following topics:

- NBWSS logging
- NBWSS issues

NBWSS logging

For messages about the NetBackup WebSocket Service (NBWSS) operations, see the following NetBackup log directories.

Table 3-1   NetBackup logs for NBWSS

Log directory	Contains the messages on	Resides on
Windows *install_path*\NetBackup\logs\nbwebservice UNIX, Linux /usr/openv/logs/nbwebservice nbwebservice uses unified logging: originator ID 485. See the *NetBackup Logging Reference Guide* for information on how to use unified logs.	Adding NetBackup endpoint credentials, and NBWSS interactions with the cloud application.	NetBackup master server

To create other NetBackup log directories

◆ Run the following command on the NetBackup servers:

Windows:

`install_path\NetBackup\logs\mklogdir.bat`

UNIX, Linux:

`/usr/openv/netbackup/logs/mklogdir`

For guidance on using NetBackup logging, see the *NetBackup Logging Reference Guide* available from the following location:

https://www.veritas.com/support/en_US/article.DOC5332

NBWSS issues

The following topics provide help in troubleshooting NBWSS and the NetBackup **WebSocket Server** dialog.

Problems validating the endpoint server in the WebSocket Server dialog

This topic describes the problems that may occur when you click **Validate** on the NetBackup **WebSocket Server** dialog to save NetBackup credentials for an endpoint.

Problems with endpoint details in a JSON-formatted file

Table 3-2   Problems adding the endpoint details from a JSON-formatted file

Error	Explanation and recommended action
`JSON contents not valid`	The endpoint information in the JSON file is invalid. For example: one or more of the fields in the JSON file are empty or contain unsupported characters. Note that NetBackup does not support an apostrophe (') anywhere in the file.
	See "WebSocket endpoint details and their formatting" on page 30.
	Correct the JSON file accordingly.

`Invalid websocket protocol. Only wss protocol supported`
 	
Or
 	
`Malformed URL:`
	The WebSocket URL in the JSON file is not in the supported format.

	Specify the URL as described in the table in the following topic:
	See "WebSocket endpoint details and their formatting" on page 30.

Table 3-2 Problems adding the endpoint details from a JSON-formatted file
(continued)

Error	Explanation and recommended action
`Unable to establish connection with host: <WebSocket servername>`	The server details are incorrect or there is a networking problem. ■ Make sure the WebSocket server's host name (or IP address) and port are correct. ■ Make sure you can ping the WebSocket server. ■ Verify that DNS lookup works.

Problems with endpoint details in a URL

Table 3-3 Problems adding the endpoint details from a URL

Error	Explanation and recommended action

`Invalid command parameter`
Or
`Malformed URL:`
	The WebSocket URL is not in the supported format.
	Specify the URL as described in the table in the following topic:
	See "WebSocket endpoint details and their formatting" on page 30.

`Failed to open connection to the remote object referred to by the URL`	NetBackup was unable to get the SSL certificate from the endpoint URL. Make sure the WebSocket server has a valid SSL certificate.
`Unable to establish connection with host: <Websocket servername>`	The server details are incorrect or there is a networking problem. ■ Make sure the WebSocket server's host name (or IP address) and port are correct. ■ Make sure you can ping the WebSocket server. ■ Verify that DNS lookup works.
`InvalidPacketException Unable to parse JSON contents`	The data that is hosted on the endpoint URL does not match the format in the table in the following topic: See "WebSocket endpoint details and their formatting" on page 30.

Problems saving the NetBackup endpoint credentials in the WebSocket Server dialog

This topic describes the problems that may occur when you click **Add Host** on the NetBackup **WebSocket Server** dialog to save NetBackup credentials for an endpoint.

Table 3-4   Problems saving the endpoint details as NetBackup credentials

Error	Explanation and recommended action
`JSON contents not valid`	The endpoint information in the JSON file is invalid. For example: one or more of the fields in the JSON file are empty or contain unsupported characters. Note that NetBackup does not support an apostrophe (') anywhere in the file. See "WebSocket endpoint details and their formatting" on page 30. Correct the JSON file accordingly.
`Invalid websocket protocol. Only wss protocol supported`	The WebSocket URL in the JSON file is not in the supported format. Specify the URL as described in the table in the following topic: See "WebSocket endpoint details and their formatting" on page 30.

`Communication with EMM failed`
 	
Or
 	
`Unable to establish connection with host: <Websocket servername>`
	The server details are incorrect or there is a networking problem.
	
	■ Make sure the WebSocket server's host name (or IP address) and port are correct.
	■ Make sure you can ping the WebSocket server.
	■ Verify that DNS lookup works.

`Webservices unable to connect to EMM. Hint: check your security settings; Config WebServices are not compatible with NBAC`	NetBackup Access Control (NBAC) mode is enabled. The `Config Webservices` do not support your current NBAC settings. Review the NBAC settings. Consider disabling NBAC.
`the entity already exists`	Make sure that an endpoint server with the same name has not already been added in NetBackup. To display the saved endpoints, click the refresh option in the **Administration Console** toolbar: Contact Veritas Technical Support for further assistance.
`The certificate did not match with the one accepted by the user, please verify the certificate`	The SSL certificate that you accepted with the **Validate** option on the **WebSocket Server** dialog does not match the certificate that the URL obtained when you clicked **Add Host**. Make sure that the SSL certificate on the endpoint server was not changed after you clicked **Validate** to accept the certificate.

Table 3-4   Problems saving the endpoint details as NetBackup credentials
(continued)

Error	Explanation and recommended action

`Failed to setup SSL security`
 	
Or
 	
`Failed to open connection to the remote object referred to by the URL`
	NetBackup was unable to get the SSL certificate from the endpoint URL.
	
	Make sure the WebSocket server has a valid SSL certificate.

`Problem occurred while storing the SSL certificate in the truststore`
 	
Or
 	
`Error loading keystore`
	NetBackup was unable to save the SSL certificate of the endpoint server to the NetBackup trust store.
	
	Contact Veritas Technical Support for further assistance.

Problems deleting the WebSocket server endpoint from NetBackup

This topic describes the problems that may occur when you delete endpoint credentials from the **WebSocket Server(s)** pane in the NetBackup Administration Console.

Table 3-5   Problems deleting the NetBackup endpoint credentials

Error	Explanation and recommended action

`Failed to remove certificate for host: <Websocket servername>`
 	
Or
 	
`Error loading keystore`
	NetBackup was unable to delete the SSL certificate of the endpoint server from the NBWSS trust store.
	
	Contact Veritas Technical Support for further assistance.

Problems displaying the list of WebSocket servers that were added in NetBackup

This topic describes the problems that may occur when you click **Media and Device Management > Credentials > WebSocket Servers** in the NetBackup Administration Console. The endpoints that have been added should appear in the **WebSocket Server(s)** pane.

Table 3-6   Problems getting the list of WebSocket server endpoints that
were added in NetBackup

Error	Explanation and recommended action
`no entity was found`	NetBackup was unable to obtain the WebSocket server endpoints, or was unable to successfully obtain information about a specific endpoint.
	Contact Veritas Technical Support for further assistance.
`Webservices unable to connect to EMM. Hint: check your security settings; Config WebServices are not compatible with NBAC`	NetBackup Access Control (NBAC) mode is enabled. The `Config Webservices` do not support your current NBAC settings.
	Review the NBAC settings. Consider disabling NBAC.

Problems activating or deactivating the endpoint server

This topic describes the problems that may occur when you attempt to activate or
deactivate the endpoint server on the **WebSocket Servers** pane of the
Administration Console.

Table 3-7   Problems activating or deactivating the endpoint server

Error	Explanation and recommended action
`Unable to establish connection with host: <Websocket servername>`	The server details are incorrect or there is a networking problem.
	■ Make sure the WebSocket server's host name (or IP address) and port are correct.
	■ Make sure you can ping the WebSocket server.
	■ Verify that DNS lookup works.

Additional NBWSS issues

This topic describes some additional NetBackup WebSocket Service (NBWSS)
problems.

Table 3-8   Additional troubleshooting issues

Problem	Recommended action
The WebSocket server's **Connection State** is `Disconnected`	Verify the following: ■ The WebSocket server is running. ■ The WebSocket server's `CONNECT RESPONSE` message contains valid information. See "NetBackup requests a connection to the endpoint" on page 16. ■ The NetBackup Web Management Console service is running.
Notifications are not sent	Verify the following: ■ The WebSocket server is running. ■ The WebSocket server's **State** is `Activated` and its **Connection State** is `Connected`. ■ The NetBackup Web Management Console service is running.