



Web services proxy

E-Series Systems

NetApp
May 25, 2022

Table of Contents

- Web services proxy 1
 - SANtricity Web Services Proxy overview 1
 - Learn about Web Services 1
 - Install and configure 9
 - Manage user access in Web Services Proxy 19
 - Manage security and certificates in Web Services Proxy 23
 - Manage storage systems using Web Services Proxy 25
 - Manage automatic polling for Web Services Proxy statistics 30
 - Manage AutoSupport using Web Services Proxy 32

Web services proxy

SANtricity Web Services Proxy overview

The SANtricity Web Services Proxy is a RESTful API server installed separately on a host system to manage hundreds of new and legacy NetApp E-Series storage systems. The proxy includes SANtricity Unified Manager, which is a web-based interface that provides similar functions.

Installation overview

Installing and configuring the Web Services Proxy involves the following steps:

1. [Review installation and upgrade requirements.](#)
2. [Download and install Web Services Proxy file.](#)
3. [Log in to API and Unified Manager.](#)
4. [Configure Web Services Proxy.](#)

Find more information

- Unified Manager — The proxy installation includes SANtricity Unified Manager, a web-based interface that provides configuration access to newer E-Series and EF-Series storage systems. For more information, see the Unified Manager online help, which is available from its user interface or from the [SANtricity software doc site](#).
- GitHub repository — GitHub contains a repository for the collection and organization of sample scripts illustrating the use of the NetApp SANtricity Web Services API. To access the repository, see [NetApp Webservices samples](#).
- Representational state transfer (REST) — Web Services is a RESTful API that provides access to virtually all the SANtricity management capabilities, so you should be familiar with REST concepts. For more information, see [Architectural Styles and the Design of Network-based Software Architectures](#).
- JavaScript Object Notation (JSON) — Because data within Web Services is encoded through JSON, you should be familiar with JSON programming concepts. For more information, see [Introducing JSON](#).

Learn about Web Services

Web Services and Unified Manager overview

Before you install and configure the Web Services proxy, read the overview of Web Services and SANtricity Unified Manager.

Web Services

Web Services is an Application Programming Interface (API) that allows you to configure, manage, and monitor NetApp E-Series and EF-Series storage systems. By issuing API requests, you can complete workflows such as configuration, provisioning, and performance monitoring for E-Series storage systems.

When using the Web Services API to manage storage systems, you should be familiar with the following:

- JavaScript Object Notation (JSON) – Because data within Web Services is encoded through JSON, you should be familiar with JSON programming concepts. For more information, see [Introducing JSON](#).
- Representational state transfer (REST) – Web Services is a RESTful API that provides access to virtually all the SANtricity management capabilities, so you should be familiar with REST concepts. For more information, see [Architectural Styles and the Design of Network-based Software Architectures](#).
- Programming language concepts – Java and Python are the most common programming languages used with the Web Services API, but any programming language that can make HTTP requests is sufficient for API interaction.

Web Services is available in two implementations:

- **Embedded** — A RESTful API server is embedded on each controller of an E2800/EF280 storage system running NetApp SANtricity 11.30 or later versions, an E5700/EF570 running SANtricity 11.40 or later versions, and an EF300 or EF600 running SANtricity 11.60 or later versions. No installation is required.
- **Proxy** — The SANtricity Web Services Proxy is a RESTful API server installed separately on a Windows or Linux server. This host-based application can manage hundreds of new and legacy NetApp E-Series storage systems. In general, you should use the proxy for networks with more than 10 storage systems. The proxy can handle numerous requests more efficiently than the embedded API.

The core of the API is available in both implementations.



The following table provides a comparison of the proxy and the embedded version.

| Consideration | Proxy | Embedded |
|---------------|--|--|
| Installation | Requires a host system (Linux or Windows). The proxy is available for download at the NetApp Support Site or on DockerHub . | No installation or enablement required. |
| Security | Minimal security settings by default. Security settings are low so that developers can get started with the API quickly and easily. If desired, you can configure the proxy with the same security profile as the embedded version. | High security settings by default. Security settings are high because the API runs directly on the controllers. For example, it does not allow HTTP access, and it disables all SSL and older TLS encryption protocols for HTTPS. |

| Consideration | Proxy | Embedded |
|--------------------|--|--|
| Central management | Manages all storage systems from one server. | Manages only the controller on which it is embedded. |

Unified Manager

The proxy installation package includes Unified Manager, a web-based interface that provides configuration access to newer E-Series and EF-Series storage systems, such as the E2800, E5700, EF300, and EF600.

From Unified Manager, you can perform the following batch operations:

- View the status of multiple storage systems from a central view
- Discover multiple storage systems in your network
- Import settings from one storage system to multiple systems
- Upgrade firmware for multiple storage systems

Compatibility and restrictions

The following compatibility and restrictions apply to using the Web Services Proxy.

| Consideration | Compatibility or restriction |
|------------------------------|--|
| HTTP support | The Web Services Proxy allows use of HTTP or HTTPS. (The embedded version of Web Services requires HTTPS for security reasons.) |
| Storage systems and firmware | The Web Services Proxy can manage all E-Series storage systems, including a mixture of older systems and the latest E2800, EF280, E5700, EF570, EF300, and EF600 series systems. |
| IP Support | <p>The Web Services Proxy supports either the IPv4 protocol or IPv6 protocol.</p> <div>  <p>The IPv6 protocol might fail when the Web Services Proxy tries to automatically discover the management address from the controller configuration. Possible causes for the failure include problems during IP address forwarding or IPv6 being enabled on the storage systems but not on the server.</p> </div> |

| Consideration | Compatibility or restriction |
|------------------------------|--|
| NVSRAM file name constraints | The Web Services Proxy uses NVSRAM file names to identify version information accurately. Therefore, you cannot change NVSRAM filenames when they are used with the Web Services Proxy. The Web Services Proxy might not recognize a renamed NVSRAM file as a valid firmware file. |
| Symbol Web | <p>Symbol Web is a URL in the REST API. It provides access to almost all symbol calls. The symbol function is part of the following URL:</p> <pre>http://host:port/devmgr/storage-system/storage array ID/symbol/symbol function</pre> <div>  <p>Symbol-disabled storage systems are supported through the Web Services Proxy.</p> </div> |

API basics

In the Web Services API, HTTP communications involve a request-response cycle.

URL elements in requests

Regardless of the programming language or tool used, each call to the Web Services API has a similar structure, with a URL, HTTP verb, and an Accept header.



All requests include a URL, as in the following example, and contain the elements described in the table.

```
https://webservices.name.com:8443/devmgr/v2/storage-systems
```

| Area | Description |
|----------------|--|
| HTTP transport | The Web Services Proxy enables the use of HTTP or HTTPS. |
| https:// | The embedded Web Services requires HTTPS for security reasons. |

| Area | Description |
|--|---|
| <p>Base URL and port</p> <p><code>webservices.name.com:8443</code></p> | <p>Each request must be correctly routed to an active instance of Web Services. The FQDN (fully qualified domain name) or the IP address of the instance is required, along with the listening port. By default, Web Services communicates over port 8080 (for HTTP) and port 8443 (for HTTPS).</p> <p>For the Web Services Proxy, both ports can be changed during the proxy installation or in the <code>wsconfig.xml</code> file. Port contention is common on data center hosts running various management applications.</p> <p>For the embedded Web Services, the port on the controller cannot be changed; it defaults to port 8443 for secure connections.</p> |
| <p>API path</p> <p><code>devmgr/v2/storage-systems</code></p> | <p>A request is made to a specific REST resource or endpoint within the Web Services API. Most endpoints are in the form of:</p> <p><code>devmgr/v2/<resource>/[id]</code></p> <p>The API path consists of three parts:</p> <ul style="list-style-type: none"> • <code>devmgr</code> (Device Manager) is the namespace of the Web Services API. • <code>v2</code> denotes the version of the API that you are accessing. You can also use <code>utils</code> to access login endpoints. • <code>storage-systems</code> is a category within the documentation. |

Supported HTTP verbs

Supported HTTP verbs include GET, POST, and DELETE:

- GET requests are used for read-only requests.
- POST requests are used to create and update objects, and also for read requests that might have security implications.
- DELETE requests are typically used to remove an object from management, remove an object entirely, or to reset the state of the object.



Currently, the Web Services API does not support PUT or PATCH. Instead, you can use POST to provide the typical functionality for these verbs.

Accept headers

When returning a request body, Web Services returns the data in JSON format (unless otherwise specified). Certain clients default to requesting “text/html” or something similar. In these cases, the API responds with an HTTP code 406, denoting that it cannot provide data in this format. As a best practice, you should define the Accept header as “application/json” for any cases in which you expect JSON as the response type. In other cases where a response body is not returned (for example, DELETE), providing the Accept header does not cause any unintended effects.

Responses

When a request is made to the API, a response returns two critical pieces of information:

- HTTP status code — Indicates whether the request was successful.
- Optional response body — Usually provides a JSON body representing the state of the resource or a body providing more details on the nature of a failure.

You must check the status code and the content-type header to determine what the resulting response body looks like. For HTTP status codes 200-203 and 422, Web Services returns a JSON body with the response. For other HTTP status codes, Web Services generally does not return an additional JSON body, either because the specification does not allow it (204) or because the status is self-explanatory. The table lists common HTTP status codes and definitions. It also indicates whether information associated with each HTTP code is returned in a JSON body.

| HTTP status code | Description | JSON body |
|-----------------------------------|---|-----------|
| 200 OK | Denotes a successful response. | Yes |
| 201 Created | Indicates that an object was created. This code is used in a few rare cases instead of a 200 status. | Yes |
| 202 Accepted | Indicates that the request is accepted for processing as an asynchronous request, but you must make a subsequent request to get the actual result. | Yes |
| 203 Non-Authoritative Information | Similar to a 200 response, but Web Services cannot guarantee that the data is up-to-date (for example, only cached data is available at this time). | Yes |
| 204 No Content | Indicates a successful operation, but there is no response body. | No |
| 400 Bad Request | Indicates that the JSON body provided in the request is not valid. | No |

| HTTP status code | Description | JSON body |
|--------------------------|---|-----------|
| 401 Unauthorized | Indicates that an authentication failure has occurred. Either no credentials were provided, or the username or password was invalid. | No |
| 403 Forbidden | An authorization failure, which indicates that the authenticated user does not have permission to access the requested endpoint. | No |
| 404 Not Found | Indicates that the requested resource could not be located. This code is valid for nonexistent APIs or nonexistent resources requested by the identifier. | No |
| 422 Unprocessable Entity | Indicates the request is generally well-formed, but either the input parameters are invalid, or the state of the storage system does not allow Web Services to satisfy the request. | Yes |
| 424 Failed Dependency | Used in the Web Services Proxy to indicate that the requested storage system is currently inaccessible. Therefore, Web Services cannot satisfy the request. | No |
| 429 Too Many Requests | Indicates that a request limit was exceeded and should be retried at a later time. | No |

Sample scripts

GitHub contains a repository for the collection and organization of sample scripts illustrating the use of the NetApp SANtricity Web Services API. To access the repository, see [NetApp Webservices samples](#).

Terms and concepts

The following terms apply to the Web Services Proxy.

| Term | Definition |
|------|--|
| API | An Application Programming Interface (API) is a set of protocols and methods that enables developers to communicate with devices. The Web Services API is used to communicate with E-Series storage systems. |

| Term | Definition |
|----------------|---|
| ASUP | The AutoSupport (ASUP) feature collects data in a customer support bundle and automatically sends the message file to technical support for remote troubleshooting and problem analysis. |
| Endpoint | Endpoints are functions that are available through the API. An endpoint includes an HTTP verb, plus the URI path. In Web Services, endpoints can execute such tasks as discovering storage systems and creating volumes. |
| HTTP Verb | An HTTP verb is a corresponding action for an endpoint, such as retrieving and creating data. In Web Services, HTTP verbs include POST, GET, and DELETE. |
| JSON | JavaScript Object Notation (JSON) is a structured data format much like XML, which uses a minimal, readable format. Data within Web Services is encoded through JSON. |
| REST / RESTful | <p>Representational state transfer (REST) is a loose specification that defines an architectural style for an API. Because most REST APIs do not fully adhere to the specification, they are described as “RESTful” or “REST-like.” Generally, a “RESTful” API is agnostic to programming languages and has the following characteristics:</p> <ul style="list-style-type: none"> • HTTP-based, which follows the general semantics of the protocol • Producer and consumer of structured data (JSON, XML, etc.) • Object-oriented (as opposed to operation-oriented) <p>Web Services is a RESTful API that provides access to virtually all the SANtricity management capabilities.</p> |
| storage system | A storage system is an E-Series array, which includes shelves, controllers, drives, software, and firmware. |
| SYMbol API | SYMbol is a legacy API for managing E-Series storage systems. The underlying implementation of the Web Services API uses SYMbol. |

| Term | Definition |
|--------------|--|
| Web Services | Web Services is an API that NetApp designed for developers to manage E-Series storage systems. There are two implementations of Web Services: embedded on the controller and a separate proxy that can be installed on Linux or Windows. |

Install and configure

Review installation and upgrade requirements

Before installing the Web Services Proxy, review the installation requirements and upgrade considerations.

Installation requirements

You can install and configure the Web Services Proxy on a Windows or Linux host system.

Proxy installation includes the following requirements.

| Requirement | Description |
|----------------------------------|---|
| Hostname limitations | Be sure that the hostname of the server where you plan to install the Web Services Proxy contains only ASCII letters, numerical digits, and hyphens (-). This requirement is due to a limitation of Java Keytool, which is used in generating a self-signed certificate for the server. If the hostname of your server contains any other characters, such as an underscore (_), the Webserver will fail to start after installation. |
| Operating systems | <p>You can install the proxy on the following operating systems:</p> <ul style="list-style-type: none"> • Linux • Windows <p>For a complete list of operating systems and firmware compatibility, see the NetApp Interoperability Matrix Tool.</p> |
| Linux: Additional Considerations | Linux Standard Base libraries (init-functions) are required for the Webserver to function properly. You must install the lsb/insserv packages for your operating system. For more information, refer to the "Additional packages required" section of the Readme file. |

| Requirement | Description |
|--------------------|---|
| Multiple instances | You can install only one instance of Web Services Proxy on a server; however, you can install the proxy on multiple servers within your network. |
| Capacity planning | <p>Web Services Proxy requires adequate space for logging. Make sure that your system meets the following available disk space requirements:</p> <ul style="list-style-type: none"> • Required installation space — 275 MB • Minimum logging space — 200 MB • System memory — 2 GB; heap space is 1 Gb by default <p>You can use a disk-space monitoring tool to verify available disk drive space for persistent storage and logging.</p> |
| License | The Web Services Proxy is a free, standalone product that does not require a license key. However, applicable copyrights and terms of service apply. If you are installing the proxy in either Graphical or Console mode, you must accept the End User License Agreement (EULA). |

Upgrade considerations

If you are upgrading from a previous version, be aware that some items are preserved or removed.

- For the Web Services Proxy, previous configuration settings are preserved. These settings include user passwords, all discovered storage systems, server certificates, trusted certificates, and server runtime configuration.
- For Unified Manager, all SANtricity OS files previously loaded in the repository are removed during the upgrade.

Download and install Web Services Proxy file

Installation involves downloading the file and then installing the proxy package on a Linux or Windows server.

Download Web Services Proxy files

You can download the installation file and the readme file from the Software download page of the NetApp Support site.

The download package includes the Web Services Proxy and the Unified Manager interface.

Steps

1. Go to [NetApp Support - Downloads](#).

2. Select **E-Series SANtricity Web Services Proxy**.
3. Follow the instructions to download the file. Make sure you select the correct download package for your server (for example, EXE for Windows; BIN or RPM for Linux).
4. Download the installation file to the server where you want to install the proxy and Unified Manager.

Install on Windows or Linux server

You can install the Web Services Proxy and Unified Manager using one of three modes (Graphical, Console, or Silent), or by using an RPM file (Linux only).

Before you begin

- [Review installation requirements](#).
- Make sure you have downloaded the correct installation file (EXE for Windows; BIN for Linux) to the server where you want to install the proxy and Unified Manager.

Graphical mode install

You can run the installation in Graphical mode for either Windows or Linux. In Graphical mode, the prompts appear in a Windows-style interface.

Steps

1. Access the folder where you downloaded the installation file.
2. Launch the installation for either Windows or Linux, as follows:

- Windows — Double-click the installation file:

```
santricity_webservices-windows_x64-nn.nn.nn.nnnn.exe
```

- Linux — Run the following command: `santricity_webservices-linux_x64-nn.nn.nn.nnnn.bin`

In the above filenames, `nn.nn.nn.nnnn` represents the version number.

The installation process starts and the NetApp SANtricity Web Services Proxy + Unified Manager splash screen appears.

3. Follow the on-screen prompts.

During the installation, you are prompted to enable several features and enter some configuration parameters. If necessary, you can change any of these selections later in the configuration files.

4. When the Webserver Started message appears, click **OK** to complete the installation.

The Install Complete dialog box appears.

5. Click the check boxes if you want to launch Unified Manager or the interactive API documentation, and then click **Done**.

Console mode install

You can run the installation in Console mode for either Windows or Linux. In Console mode, the prompts appear in the terminal window.

Steps

1. Run the following command: `<install filename> -i console`

In the above command, `<install filename>` represents the name of the proxy installation file you downloaded (for example: `santricity_webservices-windows_x64-nn.nn.nn.nnnn.exe`).



To cancel the installation at any time during the installation process, type `QUIT` at the command prompt.

The installation process starts and the Launching Installer — Introduction message appears.

2. Follow the on-screen prompts.

During the installation, you are prompted to enable several features and enter some configuration parameters. If necessary, you can change any of these selections later in the configuration files.

3. When the installation is complete, press **Enter** to exit the installer.

Silent mode install

You can run the installation in Silent mode for either Windows or Linux. In Silent mode, no return messages or scripts appear in the terminal window.

Steps

1. Run the following command: `<install filename> -i silent`

In the above command, `<install filename>` represents the name of the proxy installation file you downloaded (for example: `santricity_webservices-windows_x64-nn.nn.nn.nnnn.exe`).

2. Press **Enter**.

The installation process can take several minutes to complete. After successful installation, a command prompt appears in the terminal window.

RPM command install (Linux only)

For Linux systems that are compatible with the RPM package management system, you can install the Web Services Proxy using an optional RPM file.

Steps

1. Download the RPM file to the server where you want to install the proxy and Unified Manager.
2. Open a terminal window.
3. Enter the following command:

```
rpm -u santricity_webservices-nn.nn.nn.nnnn-n.x86_64.rpm
```



In the above command, `nn.nn.nn.nnnn` represents the version number.

The installation process can take several minutes to complete. After successful installation, a command prompt appears in the terminal window.

Log in to API and Unified Manager

Web Services includes API documentation, which enables you to directly interact with the REST API. It also includes Unified Manager, a browser-based interface for managing multiple E-Series storage systems.

Log in to Web Services API

After you install the Web Services Proxy, you can access the interactive API documentation in a browser.

The API documentation runs with each instance of Web Services, and is also available in a static PDF format from the NetApp Support site. To access the interactive version, you open a browser and enter the URL pointing to where Web Services resides (either a controller for the embedded version or a server for the proxy).



The Web Services API implements the OpenAPI specification (originally called the Swagger specification).

For initial login, you use the "admin" credentials. "Admin" is considered a super administrator with access to all functions and roles.

Steps

1. Open a browser.
2. Enter the URL for the embedded or proxy implementation:
 - Embedded: `https://<controller>:<port>/devmgr/docs/`

In this URL, `<controller>` is the IP address or FQDN of the controller, and `<port>` is the management port number of the controller (defaults to 8443).

- Proxy: `http[s]://<server>:<port>/devmgr/docs/`

In this URL, `<server>` is the IP address or FQDN of the server where the proxy is installed, and `<port>` is the listening port number (defaults to 8080 for HTTP or 8443 for HTTPS).



If the listening port is already in use, the proxy detects the conflict and prompts you to choose a different listening port.

The API documentation opens in the browser.

3. When the interactive API documentation opens, go to the drop-down menu in the upper right of the page and select **utils**.
4. Click the **Login** category to see the available endpoints.
5. Click the **POST: /login** endpoint, and then click **Try it out**.
6. For first-time login, enter admin for the username and password.
7. Click **Execute**.
8. To access the endpoints for storage management, go to the drop-down menu in the upper right and select **v2**.

The high-level categories for endpoints are displayed. You can navigate the API documentation as described in the table.

| Area | Description |
|----------------|--|
| Drop-down menu | <p>At the upper right of the page, a drop-down menu provides options for switching between version 2 of the API documentation (V2), the SYMBol interface (SYMBol V2), and API utilities (utils) for logging in.</p> <div>  <p>Because version 1 of the API documentation was a prerelease and not generally available, V1 is not included in the drop-down menu.</p> </div> |
| Categories | The API documentation is organized by high-level categories (for example: Administration, Configuration). Click on a category to see the related endpoints. |
| Endpoints | Select an endpoint to see its URL paths, required parameters, response bodies, and status codes that the URLs are likely to return. |
| Try It Out | <p>Interact with the endpoint directly by clicking Try It Out. This button is provided in each of the expanded views for endpoints.</p> <p>When you click the button, fields appear for entering parameters (if applicable). You can then enter values and click Execute.</p> <p>The interactive documentation uses JavaScript to make the request directly to the API; it is not a test request.</p> |

Log in to Unified Manager

After you install the Web Services Proxy, you can access Unified Manager to manage multiple storage systems in a web-based interface.

To access Unified Manager, you open a browser and enter the URL pointing to where the proxy is installed. The following browsers and versions are supported.

| Browser | Minimum version |
|-----------------------------|-----------------|
| Google Chrome | 79 |
| Microsoft Internet Explorer | 11 |
| Microsoft Edge | 79 |

| Browser | Minimum version |
|-----------------|-----------------|
| Mozilla Firefox | 70 |
| Safari | 12 |

Steps

1. Open a browser and enter the following URL:

```
http[s]://<server>:<port>/um
```

In this URL, `<server>` represents the IP address or FQDN of the server where the Web Services Proxy is installed, and `<port>` represents the listening port number (defaults to 8080 for HTTP or 8443 for HTTPS).

The Unified Manager login page opens.

2. For first-time login, enter `admin` for the user name, and then set and confirm a password for the admin user.

The password can include up to 30 characters. For further information about users and passwords, see the Access Management section of the Unified Manager online help.

Configure Web Services Proxy

You can modify the Web Services Proxy settings to meet the unique operating and performance requirements for your environment.

Stop or restart the Webserver

The Webserver service is started during installation and runs in the background. During some configuration tasks, you might need to stop or restart the Webserver service.

Steps

1. Do one of the following:
 - For Windows, go to the **Start** menu, select **Administrative Tools** > **Services**, locate **NetApp SANtricity Web Services** and then select either **Stop** or **Restart**.
 - For Linux, choose the method of stopping and restarting the Webserver for your operating system version. During the installation, a popup dialog indicated what daemon started. For example:

```
web_services_proxy webserver installed and started. You can interact with it
using systemctl start|stop|restart|status web_services_proxy.service
```

The most common method for interacting with the service is by using `systemctl` commands.

Resolve port conflicts

If the Web Services Proxy is running while another application is available at the defined address or port, you can resolve the port conflict in the `wsconfig.xml` file.

Steps

1. Open the wsconfig.xml file, located at:
 - (Windows) — C:\Program Files\NetApp\SANtricity Web Services Proxy
 - (Linux) — /opt/netapp/santricity_web_services_proxy
2. Add the following line to the wsconfig.xml file, in which *n* is the port number:

```
<sslport clientauth="request">*n*</sslport>
<port>n</port>
```

The following table shows the attributes that control HTTP ports and HTTPS ports.

| Name | Description | Parent Node | Attributes | Required |
|---------|--|-------------|--|----------|
| config | The root node for the config | Null | Version - The version of the config schema is currently 1.0. | Yes |
| sslport | The TCP port to listen for SSL requests. Defaults to 8443. | config | Clientauth | No |
| port | The TCP port to listen for HTTP request, defaults to 8080. | config | - | No |

3. Save and close the file.
4. Restart the Webserver service so the change takes effect.

Configure load-balancing and/or high-availability

To use the Web Services Proxy in a highly-available (HA) configuration, you can configure load balancing. In an HA configuration, typically either a single node receives all requests while the others are on stand-by, or requests are load-balanced across all nodes.

The Web Services Proxy can exist in a highly-available (HA) environment, with most APIs operating correctly regardless of the recipient of the request. Metadata tags and folders are two exceptions, because tags and folders are stored in a local database and are not shared between Web Services Proxy instances.

However, there are some known timing issues that occur in a small percentage of requests. Specifically, one instance of the proxy can have newer data faster than a second instance for a small window. The Web Services Proxy includes a special configuration that removes this timing issue. This option is not enabled by default, because it increases the amount of time it takes to service requests (for data consistency). To enable this option, you must add a property to an .INI file (for Windows) or an .SH file (for Linux).

Steps

1. Do one of the following:

- Windows: Open the appserver64.ini file, and then add the `Dload-balance.enabled=true` property.

For example: `vmarg.7=-Dload-balance.enabled=true`

- Linux: Open the webserver.sh file, and then add the `Dload-balance.enabled=true` property.

For example: `DEBUG_START_OPTIONS="-Dload-balance.enabled=true"`

2. Save your changes.
3. Restart the Webserver service so the change takes effect.

Disable SYMbol HTTPS

You can disable SYMbol commands (default setting) and send commands over a remote procedure call (RPC). This setting can be changed in the wsconfig.xml file.

By default, the Web Services Proxy sends SYMbol commands over HTTPS for all E2800 series and E5700 series storage systems running SANtricity OS versions 08.40 or later. SYMbol commands sent over HTTPS are authenticated to the storage system. If needed, you can disable HTTPS SYMbol support and send commands over RPC. Whenever SYMbol over RPC is configured, all passive commands to the storage system are enabled without authentication.



When SYMbol over RPC is used, the Web Services Proxy cannot connect to systems with the SYMbol management port disabled.

Steps

1. Open the wsconfig.xml file, located at:
 - (Windows) — `C:\Program Files\NetApp\SANtricity Web Services Proxy`
 - (Linux) — `/opt/netapp/santricity_web_services_proxy`
2. In the `devicemgt.symbolclientstrategy` entry, replace the `httpsPreferred` value with `rpcOnly`.

For example:

```
<env key="devicemgt.symbolclientstrategy">rpcOnly</env>
```

3. Save the file.

Configure cross-origin resource sharing

You can configure cross-origin resource sharing (CORS), which is a mechanism that uses additional HTTP headers to provide a web application running at one origin to have permission to access selected resources from a server at a different origin.

CORS is handled by the cors.cfg file located in the working directory. The CORS configuration is open by default, so cross domain access is not restricted.

If no configuration file is present, CORS is open. But if the cors.cfg file is present, then it is used. If the cors.cfg file is empty, you cannot make a CORS request.

Steps

1. Open the cors.cfg file, which is located in the working directory.

2. Add the desired lines to the file.

Each line in the CORS configuration file is a regular expression pattern to match. The origin header must match a line in the cors.cfg file. If any line pattern matches the origin header, the request is allowed. The complete origin is compared, not just the host element.

3. Save the file.

Requests are matched on the host and according to protocol, such as the following:

- Match localhost with any protocol — `*localhost*`
- Match localhost for HTTPS only — `https://localhost*`

Uninstall Web Services Proxy

To remove the Web Services Proxy and Unified Manager, you can use any mode (Graphical, Console, Silent, or RPM file), regardless of what method you used to install the proxy.

Graphical mode uninstall

You can run the uninstall in Graphical mode for either Windows or Linux. In Graphical mode, the prompts appear in a Windows-style interface.

Steps

1. Launch the uninstall for either Windows or Linux, as follows:
 - Windows — Go to the directory that contains the `uninstall_web_services_proxy` uninstall file. The default directory is at the following location: `C:/Program Files/NetApp/SANtricity Web Services Proxy/`. Double-click `uninstall_web_services_proxy.exe`.



Alternatively, you can go to **Control Panel > Programs > Uninstall a program**, and then select "NetApp SANtricity Web Services Proxy."

- Linux — Go to the directory that contains the Web Services Proxy uninstall file. The default directory is at the following location:
`/opt/netapp/santricity_web_services_proxy/uninstall_web_services_proxy`
2. Run the following command:

```
uninstall_web_services_proxy -i gui
```

The SANtricity Web Services Proxy splash screen appears.

3. From the Uninstall dialog box, click **Uninstall**.

The Uninstaller progress bar appears and shows the progress.

4. When the Uninstall Complete message appears, click **Done**.

Console mode uninstall

You can run the uninstall in Console mode for either Windows or Linux. In Console mode, the prompts appear

in the terminal window.

Steps

1. Go to the `uninstall_web_services_proxy` directory.
2. Run the following command:

```
uninstall_web_services_proxy -i console
```

The uninstall process starts.

3. When the uninstall is complete, press **Enter** to exit the installer.

Silent mode uninstall

You can run the uninstall in Silent mode for either Windows or Linux. In Silent mode, no return messages or scripts appear in the terminal window.

Steps

1. Go to the `uninstall_web_services_proxy` directory.
2. Run the following command:

```
uninstall_web_services_proxy -i silent
```

The uninstall process runs, but no return messages or scripts appear in the terminal window. After Web Services Proxy is successfully uninstalled, a command prompt appears in the terminal window.

RPM command uninstall (Linux only)

You can use an RPM command to uninstall the Web Services Proxy from a Linux system.

Steps

1. Open a terminal window.
2. Enter the following command line:

```
rpm -e santricity_webservices
```



The uninstall process might leave files that were not part of the original installation. Manually delete these files to remove Web Services Proxy completely.

Manage user access in Web Services Proxy

You can manage user access to the Web Services API and Unified Manager for security purposes.

Overview of access management

Access management includes role-based logins, password encryption, basic authentication, and LDAP integration.

Role-based access

Role-based access control (RBAC) associates predefined users with roles. Each role grants permissions to a specific level of functionality.

The following table describes each role.

| Role | Description |
|-----------------|--|
| security.admin | SSL and certificate management. |
| storage.admin | Full read/write access to storage system configuration. |
| storage.monitor | Read-only access to view storage system data. |
| support.admin | Access to all hardware resources on storage systems and support operations such as AutoSupport (ASUP) retrieval. |

Default user accounts are defined in the users.properties file. You can change user accounts by directly modifying the users.properties file or by using the Access Management functions in Unified Manager.

The following table lists the user logins available for the Web Services Proxy.

| Predefined user login | Description |
|-----------------------|--|
| admin | A super administrator who has access to all functions and includes all roles. For Unified Manager, you must set the password on first-time login. |
| storage | The administrator responsible for all storage provisioning. This user includes the following roles: storage.admin, support.admin, and storage.monitor. This account is disabled until a password is set. |
| security | The user responsible for security configuration. This user includes the following roles: security.admin and storage.monitor. This account is disabled until a password is set. |
| support | The user responsible for hardware resources, failure data, and firmware upgrades. This user includes the following roles: support.admin and storage.monitor. This account is disabled until a password is set. |
| monitor | A user with read-only access to the system. This user includes only the storage.monitor role. This account is disabled until a password is set. |

| Predefined user login | Description |
|-----------------------|---|
| rw | The rw (read/write) user includes the following roles: storage.admin, support.admin, and storage.monitor. This account is disabled until a password is set. |
| ro | The ro (read only) user includes only the storage.monitor role. This account is disabled until a password is set. |

Password encryption

For each password, you can apply an additional encryption process using the existing SHA256 password encoding. This additional encryption process applies a random set of bytes to each password (salt) for each SHA256 hash encryption. Salted SHA256 encryption is applied to all newly created passwords.



Prior to the Web Services Proxy 3.0 release, passwords were encrypted through SHA256 hashing only. Any existing SHA256 hash-only encrypted passwords retain this encoding and are still valid under the users.properties file. However, SHA256 hash-only encrypted passwords are not as secure as those passwords with salted SHA256 encryption.

Basic authentication

By default, basic authentication is enabled, which means the server returns a basic authentication challenge. This setting can be changed in the wsconfig.xml file.

LDAP

Lightweight Directory Access Protocol (LDAP), an application protocol for accessing and maintaining distributed directory information services, is enabled for the Web Services Proxy. LDAP integration allows for user authentication and mapping of roles to groups.

For information on configuring LDAP functionality, refer to configuration options in the Unified Manager interface or in the LDAP section of the interactive API documentation.

Configure user access

You can manage user access by applying additional encryption to passwords, setting basic authentication, and defining role-based access.

Apply additional encryption to passwords

For the highest level of security, you can apply additional encryption to passwords using the existing SHA256 password encoding.

This additional encryption process applies a random set of bytes to each password (salt) for each SHA256 hash encryption. Salted SHA256 encryption is applied to all newly created passwords.

Steps

1. Open the users.properties file, located at:
 - (Windows) — C:\Program Files\NetApp\SANtricity Web Services Proxy\data/config

- (Linux) — /opt/netapp/santricity_web_services_proxy/data/config
2. Re-enter the encrypted password as plain text.
 3. Run the `securepasswd` command line utility to re-encrypt the password or simply restart the Web Services Proxy. This utility is installed in the root install directory for the Web Services Proxy.



Alternatively, you can salt and hash local user passwords whenever password edits are performed through the Unified Manager.

Configure basic authentication

By default basic authentication is enabled, which means the server returns a basic authentication challenge. If desired, you can change that setting in the `wsconfig.xml` file.

1. Open the `wsconfig.xml` file, located at:
 - (Windows) — C:\Program Files\NetApp\SANtricity Web Services Proxy
 - (Linux) — /opt/netapp/santricity_web_services_proxy
2. Modify the following line in the file by specifying `false` (not enabled) or `true` (enabled).

For example: `<env key="enable-basic-auth">true</env>`

3. Save the file.
4. Restart the Webserver service so the change takes effect.

Configure role-based access

To limit user access to specific functions, you can modify which roles are specified for each user account.

The Web Services Proxy includes role-based access control (RBAC), in which roles are associated with predefined users. Each role grants permissions to a specific level of functionality. You can change the roles assigned to user accounts by directly modifying the `users.properties` file.



You can also change user accounts by using Access Management in Unified Manager. For more information, see the online help available with Unified Manager.

Steps

1. Open the `users.properties` file, located in:
 - (Windows) — C:\Program Files\NetApp\SANtricity Web Services Proxy\data\config
 - (Linux) — /opt/netapp/santricity_web_services_proxy/data/config
2. Locate the line for the user account you want to modify (storage, security, monitor, support, rw, or ro).



Do not modify the admin user. This is a super user with access to all functions.

3. Add or remove the specified roles, as desired.

Roles include:

- `security.admin` — SSL and certificate management.

- storage.admin — Full read/write access to storage system configuration.
- storage.monitor — Read-only access to view storage system data.
- support.admin — Access to all hardware resources on storage systems and support operations such as AutoSupport (ASUP) retrieval.



The storage.monitor role is required for all users, including the administrator.

4. Save the file.

Manage security and certificates in Web Services Proxy

For security in the Web Services Proxy, you can specify an SSL port designation and you can manage certificates. Certificates identify website owners for secure connections between clients and servers.

Enable SSL

The Web Services Proxy uses Secure Sockets Layer (SSL) for security, which is enabled during installation. You can change the SSL port designation in the wsconfig.xml file.

Steps

1. Open the wsconfig.xml file, located at:
 - (Windows) — C:\Program Files\NetApp\SANtricity Web Services Proxy
 - (Linux) — /opt/netapp/santricity_web_services_proxy
2. Add or change the SSL port number, similar to the following example:

```
<sslport clientauth="request">8443</sslport>
```

Result

When the server is started with SSL configured, the server looks for the keystore and truststore files.

- If the server does not find a keystore, the server uses the IP address of the first detected non-loopback IPv4 address to generate a keystore and then add a self-signed certificate to the keystore.
- If the server does not find a truststore, or the truststore is not specified, the server uses the keystore as the truststore.

Bypass certificate validation

To support secure connections, the Web Services Proxy validates the storage systems' certificates against its own trusted certificates. If needed, you can specify that the proxy bypass that validation before connecting to the storage systems.

Before you begin

- All storage system connections must be secure.

Steps

1. Open the wsconfig.xml file, located at:
 - (Windows) — C:\Program Files\NetApp\SANtricity Web Services Proxy
 - (Linux) — /opt/netapp/santricity_web_services_proxy
2. Enter true in the trust.all.arrays entry, as shown in the example:

```
<env key="trust.all.arrays">true</env>
```

3. Save the file.

Generate and import a host management certificate

Certificates identify website owners for secure connections between clients and servers. To generate and import Certificate Authority (CA) certificates for the host system where the Web Services Proxy is installed, you can use API endpoints.

To manage certificates for the host system, you perform the following tasks using the API:

- Create a certificate signing request (CSR) for the host system.
- Send the CSR file to a CA, and then wait for them to send you the certificate files.
- Import the signed certificates to the host system.



You can also manage certificates in the Unified Manager interface. For more information, see the online help available in Unified Manager.

Steps

1. Log in to the [interactive API documentation](#).
2. Go to the drop-down menu in the upper right and then select **v2**.
3. Expand the **Administration** link and scroll down to the **/certificates** endpoints.
4. Generate the CSR file:
 - a. Select **POST:/certificates**, and then select **Try it out**.

The web server regenerates a self-signed certificate. You can then enter information in the fields to define the common name, organization, organization unit, alternate ID, and other information used to generate the CSR.

- b. Add the required information in the **Example values** pane to generate a valid CA certificate, and then execute the commands.



Do not call **POST:/certificates** or **POST:/certificates/reset** again, or you must regenerate the CSR. When you call **POST:/certificates** or **POST:/certificates/reset**, you are generating a new self-signed certificate with a new private key. If you send a CSR that was generated before the last reset of the private key on the server, the new security certificate does not work. You must generate a new CSR and request a new CA certificate.

- c. Execute the **GET:/certificates/server** endpoint to confirm that the current certificate status is the self-signed certificate with the information added from the **POST:/certificates** command.

The server certificate (denoted by the alias `jetty`) is still self-signed at this point.

- d. Expand the **POST:/certificates/export** endpoint, select **Try it out**, enter a file name for the CSR file, and then click **Execute**.
5. Copy and paste the `fileUrl` into a new browser tab to download the CSR file, and then send the CSR file to a valid CA to request a new web server certificate chain.
6. When the CA issues a new certificate chain, use a certificate manager tool to break out the root, intermediate, and web server certificates, and then import them to the Web Services Proxy server:
 - a. Expand the **POST:/sslconfig/server** endpoint and select **Try it out**.
 - b. Enter a name for the CA root certificate in the **alias** field.
 - c. Select **false** in the **replaceMainServerCertificate** field.
 - d. Browse to and select the new CA root certificate.
 - e. Click **Execute**.
 - f. Confirm that the certificate upload was successful.
 - g. Repeat the CA certificate upload procedure for the CA intermediate certificate.
 - h. Repeat the certificate upload procedure for the new web server security certificate file, except in this step, select **true** on the **replaceMainServerCertificate** drop-down.
 - i. Confirm that the web server security certificate import was successful.
 - j. To confirm that the new root, intermediate, and web server certificates are available in the keystore, run **GET:/certificates/server**.
7. Select and expand the **POST:/certificates/reload** endpoint, and then select **Try it out**. When prompted, whether you want to restart both controllers or not, select **false**. ("True" applies only in the case of dual array controllers.) Click **Execute**.

The **/certificates/reload** endpoint usually returns a successful http 202 response. However, the reload of the web server truststore and keystore certificates does create a race condition between the API process and the web server certificate reload process. In rare cases, the web server certificate reload can beat the API processing. In this case, the reload appears to fail even though it completed successfully. If this occurs, continue to the next step anyway. If the reload actually failed, the next step also fails.

8. Close the current browser session to the Web Services Proxy, open a new browser session, and confirm that a new secure browser connection to the Web Services Proxy can be established.

By using an incognito or in-private browsing session, you can open a connection to the server without using any saved data from previous browsing sessions.

Manage storage systems using Web Services Proxy

To manage storage systems in the network, you must first discover them and then add them to the management list.

Discover storage systems

You can set automatic discovery or manually discover storage systems.

Automatically discover storage systems

You can specify that storage systems are automatically discovered in the network by modifying the settings in the `wsconfig.xml` file. By default, IPv6 automatic discovery is disabled and IPv4 is enabled.

You only need to provide one management IP or DNS address to add a storage system. The server automatically discovers all management paths when the paths are either not configured or the paths are configured and rotatable.



If you attempt to use an IPv6 protocol to automatically discover storage systems from the controller configuration after an initial connection has been made, the process might fail. Possible causes for the failure include problems during IP address forwarding or IPv6 being enabled on the storage systems, but not being enabled on the server.

Before you begin

Before enabling IPv6 discovery settings, verify that your infrastructure supports IPv6 connectivity to the storage systems to mitigate any connection issues.

Steps

1. Open the `wsconfig.xml` file, located at:
 - (Windows) — `C:\Program Files\NetApp\SANtricity Web Services Proxy`
 - (Linux) — `/opt/netapp/santricity_web_services_proxy`
2. In the autodiscover strings, change settings from `true` to `false`, as desired. See the following example.

```
<env key="autodiscover.ipv6.enable">true</env>
```



When the paths are configured, but not configured so that the server can route to the addresses, intermittent connection errors happen. If you cannot set the IP addresses to be routable from the host, turn off auto discovery (change the settings to `false`).

3. Save the file.

Discover and add storage systems using API endpoints

You can use API endpoints to discover and add storage systems to the managed list. This procedure creates a management connection between the storage system and the API.



This task describes how to discover and add storage systems using the REST API, so you can manage these systems in the interactive API documentation. However, you might want to manage storage systems in the Unified Manager instead, which provides an easy-to-use interface. For more information, see the online help available with Unified Manager.

Before you begin

For storage systems with SANtricity versions 11.30 and later, the legacy management interface for SYMBol must be enabled in the SANtricity System Manager interface. Otherwise, the Discovery endpoints fail. You can find this setting by opening System Manager, and then going to **Settings > System > Additional Settings > Change Management Interface**.

Steps

1. Log in to the [interactive API documentation](#).
2. Discover storage systems, as follows:
 - a. From the API documentation, make sure **V2** is selected in the drop-down, and then expand the **Storage-Systems** category.
 - b. Click the **POST: /discovery** endpoint, and then click **Try it out**.
 - c. Enter the parameters as described in the table.

| | |
|-------------------|--|
| startIP endIP | Replace string with the starting and ending IP address range for one or more storage systems in the network. |
| useAgents | Set this value to either: <ul style="list-style-type: none"> • true = Use in-band agents for the network scan. • false = Do not use in-band agents for the network scan. |
| connectionTimeout | Enter the seconds allowed for the scan before the connection times out. |
| maxPortsToUse | Enter a maximum number of ports used for the network scan. |

- d. Click **Execute**.



API actions execute without user prompts.

The discovery process runs in the background.

- e. Make sure the code returns a 202.
 - f. Under **Response Body**, locate the value returned for the requestId. You need the Request ID to view the results in the next step.
3. View discovery results, as follows:
 - a. Click the **GET: /discovery** endpoint, and then click **Try it out**.
 - b. Enter the Request ID from the previous step. If you leave the **Request ID** blank, the endpoint defaults to the last request ID executed.
 - c. Click **Execute**.
 - d. Make sure the code returns 200.
 - e. In the response body, locate your Request ID and the strings for storageSystems. The strings look similar to the following example:

```

"storageSystems": [
  {
    "serialNumber": "123456789",
    "wwn": "000A011000AF00000000000001A0C000E",
    "label": "EF570_Array",
    "firmware": "08.41.10.01",
    "nvsram": "N5700-841834-001",
    "ipAddresses": [
      "10.xxx.xx.213",
      "10.xxx.xx.214"
    ],
  },

```

f. Write down the values for wwn, label, and ipAddresses. You need them for the next step.

4. Add storage systems, as follows:

- a. Click the **POST: /storage-system** endpoint, and then click **Try it out**.
- b. Enter the parameters as described in the table.

| | |
|---------------------|--|
| id | Enter a unique name for this storage system. You can enter the label (displayed in the response for GET: /discovery), but the name can be any string you choose. If you do not provide a value for this field, Web Services automatically assigns a unique identifier. |
| controllerAddresses | Enter the IP addresses displayed in the response for GET: /discovery. For dual controllers, separate the IP addresses with a comma. For example: "IP address 1", "IP address 2" |
| validate | Enter <code>true</code> , so you can receive confirmation that Web Services can connect to the storage system. |
| password | Enter the administrative password for the storage system. |
| wwn | Enter the WWN of the storage system (displayed in the response for GET: /discovery). |

- c. Remove all strings after "enableTrace": `true`, so that the entire string set is similar to the following example:

```
{
  "id": "EF570_Array",
  "controllerAddresses": [
    "Controller-A-Mgmt-IP", "Controller-B-Mgmt_IP"
  ],
  "validate": true,
  "password": "array-admin-password",
  "wwn": "000A011000AF00000000000001A0C000E",
  "enableTrace": true
}
```

d. Click **Execute**.

e. Make sure the code response is 201, which indicates that the endpoint executed successfully.

The **Post: /storage-systems** endpoint is queued. You can view the results using the **GET: /storage-systems** endpoint in the next step.

5. Confirm the list addition, as follows:

a. Click the **GET: /storage-system** endpoint.

No parameters are required.

b. Click **Execute**.

c. Make sure that the code response is 200, which indicates that the endpoint executed successfully.

d. In the response body, look for the storage system details. The returned values indicate that it was successfully added to the list of managed arrays, similar to the following example:

```
[
  {
    "id": "EF570_Array",
    "name": "EF570_Array",
    "wwn": "000A011000AF00000000000001A0C000E",
    "passwordStatus": "valid",
    "passwordSet": true,
    "status": "optimal",
    "ip1": "10.xxx.xx.213",
    "ip2": "10.xxx.xx.214",
    "managementPaths": [
      "10.xxx.xx.213",
      "10.xxx.xx.214"
    ]
  }
]
```

Scale up the number of managed storage systems

By default, the API can manage up to 100 storage systems. If you need to manage more, you must bump the memory requirements for the server.

The server is set to use 512 MB of memory. For every 100 extra storage systems in your network, add 250 MB to that number. Do not add more memory than what you physically have. Allow enough extra for your operating system and other applications.



The default cache size is 8,192 events. The approximate data usage for the MEL events cache is 1MB for each 8,192 events. Therefore, by retaining the defaults, cache usage should be approximately 1MB for a storage system.



In addition to memory, the proxy uses network ports for each storage system. Linux and Windows consider network ports as file handles. As a security measure, most operating systems limit the number of open file handles that a process or a user can have open at one time. Especially in Linux environments, where open TCP connections are considered to be file handles, the Web Services Proxy can easily exceed this limit. Because the fix is system dependent, you should refer to your operating system's documentation for how to raise this value.

Steps

1. Do one of the following:
 - On Windows, go to the `appserver64.init` file. Locate the line, `vmarg.3=-Xmx512M`
 - On Linux, go to the `webserver.sh` file. Locate the line, `JAVA_OPTIONS="-Xmx512M"`
2. To increase the memory, replace 512 with the desired memory in MB.
3. Save the file.

Manage automatic polling for Web Services Proxy statistics

You can configure automatic polling for all disk and volume statistics on discovered storage systems.

Overview of statistics

Statistics provide information about the data collection rates and performance of the storage systems.

The Web Services Proxy provides access to the following types of statistics:

- Raw statistics — Total counters for data points at the time of data collection. Raw statistics can be used for total read operations or total write operations.
- Analyzed statistics — Calculated information for an interval. Examples of analyzed statistics are read input/output operations (IOPs) per second or write throughput.

Raw statistics are linear, typically requiring at least two collected data points to derive usable data from them. The analyzed statistics are a derivation of the raw statistics, which provide important metrics. Many values that can be derived from the raw statistics are shown in a usable, point-in-time format in the analyzed statistics for your convenience.

You can retrieve raw statistics regardless of whether the automatic polling is enabled or not. You can add the `usecache=true` query string to the end of the URL to retrieve cached statistics from the last poll. Using cached results greatly increases the performance of statistics retrieval. However, multiple calls at a rate equal to or less than the configured polling interval cache retrieves the same data.

Statistics functionality

The Web Services Proxy provides API endpoints that enable the retrieval of raw and analyzed controller and interface statistics from supported hardware models and software versions.

Raw Statistics APIs

- `/storage-systems/{system-id}/controller-statistics`
- `/storage-systems/{system-id}/drive-statistics/{optional list of disk ids}`
- `/storage-systems/{system-id}/interface-statistics/{optional list of interface ids}`
- `/storage-systems/{system-id}/volume-statistics/{optional list of volume ids}`

Analyzed Statistics APIs

- `/storage-systems/{id}/analysed-controller-statistics/`
- `/storage-systems/{id}/analysed-drive-statistics/{optional list of disk ids}`
- `/storage-systems/{id}/analysed-interface-statistics/{optional list of interface ids}`
- `/storage-systems/{id}/analysed-volume-statistics/{optional list of volume ids}`

These URLs retrieve analyzed statistics from the last poll and are only available when polling is enabled. These URLs include the following input-output data:

- Operations per second
- Throughput in megabytes per second
- Response times in milliseconds

The calculations are based on the differences between statistical polling iterations, which are the most common measures of storage performance. These statistics are preferable to unanalyzed statistics.



When the system starts, there is no previous statistics collection to use to calculate the various metrics, so analyzed statistics require at least one polling cycle after startup to return data. In addition, if the cumulative counters are reset, the next polling cycle will have unpredictable numbers for the data.

Configure polling intervals

To configure polling intervals, you modify the `wsconfig.xml` file to specify a polling interval in seconds.



Because the statistics are cached in memory, you might see an increase of about 1.5 MB of memory-use for each storage system.

Before you begin

- The storage systems must be discovered by the proxy.

Steps

1. Open the wsconfig.xml file, located at:
 - (Windows) — C:\Program Files\NetApp\SANtricity Web Services Proxy
 - (Linux) — /opt/netapp/santricity_web_services_proxy
2. Add the following line inside the `<env-entries>` tag, in which `n` is the number of seconds for the interval between polling requests:

```
<env key="stats.poll.interval">n</env>
```

For example, if 60 is entered, polling starts at 60-second intervals. That is, the system requests polling to start 60 seconds after the prior polling period was completed (regardless of the duration of the prior polling period). All statistics are time-stamped with the exact time they were retrieved. The system uses the time stamp or time difference on which to base the 60-second calculation.

3. Save the file.

Manage AutoSupport using Web Services Proxy

You can configure AutoSupport (ASUP), which collects data and then automatically sends that data to technical support for remote troubleshooting and problem analysis.

Overview of AutoSupport (ASUP)

The AutoSupport (ASUP) feature automatically transmits messages to NetApp based on manual and schedule-based criteria.

Each AutoSupport message is a collection of log files, configuration data, state data, and performance metrics. By default, AutoSupport transmits the files listed in the following table to the NetApp Support team once each week.

| File Name | Description |
|----------------------|---|
| x-headers-data.txt | A .txt file containing the X-header information. |
| manifest.xml | An .xml file detailing the contents of the message. |
| arraydata.xml | An .xml file containing the list of client persisted data. |
| appserver-config.txt | A .txt file containing application server configuration data. |
| wsconfig.txt | A .txt file containing the web service configuration data. |

| File Name | Description |
|--------------------------|--|
| host-info.txt | A .txt file containing information about the host environment. |
| server-logs.7z | A .7z file containing every available webserver log file. |
| client-info.txt | A .txt file with arbitrary key/value pairs for application-specific counters such as method and webpage hits. |
| webservices-profile.json | <p>These files contain Webservices profile data and Jersey monitoring statistical data. By default, Jersey monitoring statistics are enabled. You can enable and disable them in the wsconfig.xml file, as follows:</p> <ul style="list-style-type: none"> • Enable: <code><env key="enable.jersey.statistics">true</env></code> • Disable: <code><env key="enable.jersey.statistics">false</env></code> |

Configure AutoSupport

AutoSupport is enabled by default at installation; however, you can change that setting or modify the delivery types.

Enable or disable AutoSupport

The AutoSupport feature is enabled or disabled during the initial installation of the Web Services Proxy, but you can change that setting in the ASUPConfig file.

You can enable or disable AutoSupport through the ASUPConfig.xml file, as described in the steps below. Alternatively, you can enable or disable this feature through the API using **Configuration** and **POST/asup**, and then entering "true" or "false."

1. Open the ASUPConfig.xml file in the working directory.
2. Locate the lines for `<asupdata enabled="(Boolean)" timestamp=>`
3. Enter `true` (enable) or `false` (disable). For example:

```
<asupdata enabled="false" timestamp="0">
```



The timestamp entry is superfluous.

4. Save the file.

Configure AutoSupport delivery method

You can configure the AutoSupport feature to use HTTPS, HTTP, or SMTP delivery methods. HTTPS is the default delivery method.

1. Access the ASUPConfig.xml file in the working directory.
2. In the string, `<delivery type="n">`, enter 1, 2, or 3 as described in the table:

| Value | Description |
|-------|---|
| 1 | HTTPS (default) <code><delivery type="1"></code> |
| 2 | HTTP <code><delivery type="2"></code> |
| 3 | SMTP — To properly configure the AutoSupport delivery type to SMTP, you must include the SMTP mail server address, along with the sender and recipient user emails, similar to the following example: <pre><delivery type="3"> <smtp> <mailserver>smtp.example.com</mailserver> <sender>user@example.com</sender> <replyto>user@example.com</replyto> </smtp> </delivery></pre> |

Copyright Information

Copyright © 2022 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.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

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.