NETCONF Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 29, 2018

Q. Wu
Y. Niu
Huawei

June 27, 2018

Factory default Setting Capability for RESTCONF
draft-wu-netconf-restconf-factory-restore-00

Abstract

This document defines capability based extension to RESTCONF protocol that allows RESTCONF client to configure newly deployed devices with just its preconfigured initial state (i.e., factory default settings) during zero touch bootstrapping process or restore the configuration to its preconfigured initial state or system restore point either during device rooting process or at the time of system fatal error or malfunction.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

RESTCONF uses HTTP methods such as HTTP POST, PUT, PATCH, and DELETE to provide CRUD operations on a conceptual datastore containing YANG-defined data, which is compatible with a server that implements NETCONF datastores. As described in [RFC8040], the HTTP PUT method on the datastore resource can be used to replace the entire content of the Datasore, however it can not be used to return any datastore (e.g., <startup>) to factory default setting or previous system restore point,especially when the RESTCONF server is implemented in a device that does not have NETCONF support.

This document defines capability based extension to RESTCONF protocol that allows RESTCONF client to configure newly deployed devices with just its preconfigured initial state (i.e., factory default settings) during zero touch bootstrapping process or restore the configuration to its preconfigured initial state or system restore point either during device rooting process or at the time of system fatal error.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [RFC8342] and are not redefined here:

- operational state datastore

- running configuration datastore

- intended configuration datastore

2. Datastore

A RESTCONF server implementing this document MUST be NMDA-compliant [I-D.ietf-netconf-nmda-restconf] and is implemented in a device that does not have a NETCONF server [RFC8040]. A RESTCONF server supports both the operational state Datastore and the intended configuration datastore.

2.1. The factory default setting Configuration Datastore

This document introduces a new configuration datastore named 'Factory default setting' that represents datastore with its preconfigured initial state.

The Factory default setting datastore is essentially a special case of startup configuration datastore. The new name is used in order to emphasize the narrower semantics: the factory default setting datastore MUST be preconfigured with initial state of factory default setting.

The factory default setting datastore assumes the place of the datastore resource as defined in [RFC8040] Section 3.4. This means that the entire datastore resources inside the "{+restconf}/data" subtree correspond to data instances in the factory default setting datastore. Therefore, the contents of the factory default setting datastore can be retrieved by means of the GET method as specified in [RFC8040] and but can not be modified by means of PUT methods as specified in [RFC8040].

3. New Operations

In order to support factory restore and system restore capability in RESTCONF, the YANG module "ietf-restconf-restore" defines three operations below. The factory-restore operation is is similar to NETCONF <delete-config> and defined to support Factory default Setting Capability in RESTCONF.

3.1. factory-restore

The <factor-restore> operation atomically returns any target datastore to factory default setting. The <factor-restore>operation replaces the entire contents of a writable Datastore with the contents of complete factory default setting, similar to the <delete-config> operation defined in [RFC6241], but with additional flexibility in specifying the target Datastore. If the target datastore exists, it is overwritten. Otherwise, a new one is created, if allowed. If an <factory-restore> operation is invoked on a non-writable datastore, then an error is returned, as specified in"ietf-netconf-nmda".

The "source" parameter is a datastore identity that indicates the desired source of <factor-restore> operation.

The "target" parameter is a name of the configuration datastore to use as the destination of the <factory-restore> operation.

The <url> element can appear as the <source> or <target> parameter.

The "restart" parameter is used to indicate whether to use bootstrapping to return target datastore to factory default setting in the < factor-restore > operation. If bootstrapping is used, the "source" parameter MAY not be specified.

3.1.1. Example: return <running> to factory default setting without zero touch bootstrapping support

The client(e.g.,NMS) might send the following POST request message to invoke the "factory-recovery" RPC operation:

```
      POST /restconf/operations/ietf-restconf-factory-default:factory-recovery
HTTP/1.1
      Host: example.com
      Content-Type: application/yang-data+xml
     <input xmlns="https://example.com/ns/ ietf-restconf-factory-default ">
       <source>
       <url>file://checkpoint.conf</url>
       </source>
        <target>running</target>
        <restart>false</restart>
      </input>
```

In this request, <restart> element should be set to false, <source> element MUST be specified. If URL Capability is supported, <url>element in the <source> element MUST be specified. The server will use HTTP GET method to retrieve content of <source> corresponding to factory default setting datastore and copy the

entire content to <target>. In successful case, the server might respond as follows:

```
      HTTP/1.1 204 No Content
      Date: Thu, 26 Jan 2017 20:56:30 GMT
      Server: example-server
```

3.1.2. Example: return <running> to factory default setting with zero touch bootstrapping support

The client(e.g.,NMS) might send the following POST request message to invoke the "factory-recovery" RPC operation:

```
      POST /restconf/operations/ietf-restconf-factory-default:factory-recovery
HTTP/1.1
      Host: example.com
      Content-Type: application/yang-data+xml
        <input xmlns="https://example.com/ns/ ietf-restconf-factory-default ">
        <target>running</target>
        <restart>true</restart>
      </input>
```

In this request, <restart> element should be set to true, <source> element is not specified. The server will use zerotouch bootstrap service defined in [I-D.ietf-netconf-zerotouch] to get factory default setting configuration and copy the entire content to <target>. In successful case, the server might respond as follows:

```
      HTTP/1.1 204 No Content
      Date: Thu, 26 Jan 2017 20:56:30 GMT
      Server: example-server
```

4. YANG Module

```
   <CODE BEGINS> file "ietf-restconf-restore@2018-06-21.yang"
   module ietf-restconf-restore {
     namespace
       "urn:ietf:params:xml:ns:yang:ietf-restconf-restore";
     prefix rct;

    import ietf-inet-types {
      prefix inet;
    }
     organization
       "IETF NETCONF (Network Configuration) Working Group";

     contact
       "WG Web:   <https://tools.ietf.org/wg/netconf/>
        WG List:  <mailto:netconf@ietf.org>
```

```
        WG Chair: Kent Watsen
                  <mailto:kwatsen@juniper.net>

        WG Chair: Mahesh Jethanandani
                  <mailto:mjethanandani@gmail.com>

        Editor:   Qin Wu
                  <mailto:bill.wu@huawei.com>";

     description
       "This module defines operations that implement factory-default and
        system restore capability in the RESTCONF protocol.

        Copyright (c) 2018 IETF Trust and the persons identified as
        authors of the code. All rights reserved.

        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject to
        the license terms contained in, the Simplified BSD License set
       forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
        (https://trustee.ietf.org/license-info).

        The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
        NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'MAY', and
        'OPTIONAL' in the module text are to be interpreted as described
        in RFC 2119 (https://tools.ietf.org/html/rfc2119).

        This version of this YANG module is part of RFC XXXX
        (https://tools.ietf.org/html/rfcXXXX); see the RFC itself for
        full legal notices.";

     revision 2018-06-21 {
       description
         "Initial revision.";
       reference
         "RFC XXXX: Factory default Setting Capability for RESTCONF";
     }

// NETCONF capabilities defined as features
    feature startup {
      description
        "NETCONF :startup capability;
         If the server advertises the :startup
         capability for a session, then this feature must
         also be enabled for that session.  Otherwise,
         this feature must not be enabled.";
      reference "RFC 6241, Section 8.7";
```

```
    }

    feature url {
      description
        "NETCONF :url capability;
         If the server advertises the :url
         capability for a session, then this feature must
         also be enabled for that session.  Otherwise,
         this feature must not be enabled.";
      reference "RFC 6241, Section 8.8";
    }
      feature writable-running {
      description
        "NETCONF :writable-running capability;
         If the server advertises the :writable-running
         capability for a session, then this feature must
         also be enabled for that session.  Otherwise,
         this feature must not be enabled.";
      reference "RFC 6241, Section 8.2";
    }

    feature candidate {
      description
        "NETCONF :candidate capability;
         If the server advertises the :candidate
         capability for a session, then this feature must
         also be enabled for that session.  Otherwise,
         this feature must not be enabled.";
      reference "RFC 6241, Section 8.3";
    }

     /* Operations */

     rpc factory-recovery {
       description
         "The <factor-restore> operation atomically returns any target
datastore
         to factory default setting.  The <factor-restore>operation replaces
the
          entire contents of a writable Datastore with the contents of complete
         factory default setting, similar to the <delete-config> operation
defined in
         [RFC6241], but with additional flexibility in specifying the target
Datastore.
          ";

      input {
        container target {
          description
            "Particular configuration to return factory default setting to.";
          choice config-target {
            mandatory true;
            description
              "The configuration target of the copy operation.";
            leaf candidate {
              if-feature candidate;
              type empty;
              description
                "The candidate configuration is the config target.";
            }
            leaf running {
              if-feature writable-running;
              type empty;
              description
                "The running configuration is the config target.
                 This is optional-to-implement on the server.";
            }

            leaf startup {
              if-feature startup;
              type empty;
              description
                "The startup configuration is the config target.";
            }
            leaf url {
              if-feature url;
              type inet:uri;
              description
                "The URL-based configuration is the config target.";
            }
          }
        }

        container source {
          description
            "Particular configuration to copy from.";
          choice config-source {
            mandatory true;
           description
              "The configuration source for the copy operation.";

            leaf factory {
              if-feature candidate;
              type empty;
              description
                "The factory-default-setting configuration is the config
source.";
            }

            leaf url {
             if-feature url;
```

```
              type inet:uri;
              description
                "The URL-based configuration is the config source.";
            }
          }
        }
      leaf restart {
       type boolean;
            description
              "indicate whether restart or zero touch bootstrapping
               service is enabled.";
        }
      }
     }
}
   <CODE ENDS>
```

5. IANA Considerations

This document registers one URI in the IETF XML Registry [RFC3688]. The following registration has been made:

URI: urn:ietf:params:xml:ns:yang:ietf-restconf-restore

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

This document registers one YANG module in the YANG Module Names Registry [RFC6020]. The following registration has been made:

name: ietf-restconf-restore

namespace: urn:ietf:params:xml:ns:yang:ietf-restconf- restore

prefix: rcf

RFC: xxxx

6. Security Considerations

TBD.

7. Normative References

[I-D.ietf-netconf-zerotouch]
Watsen, K., Abrahamsson, M., and I. Farrer, "Zero Touch Provisioning for Networking Devices", draft-ietf-netconf-zerotouch-22 (work in progress), June 2018.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, DOI 10.17487/RFC5277, July 2008, <https://www.rfc-editor.org/info/rfc5277>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6021] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6021, DOI 10.17487/RFC6021, October 2010, <https://www.rfc-editor.org/info/rfc6021>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC6470] Bierman, A., "Network Configuration Protocol (NETCONF) Base Notifications", RFC 6470, DOI 10.17487/RFC6470, February 2012, <https://www.rfc-editor.org/info/rfc6470>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

Authors' Addresses

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Ye Niu
Huawei

Email: niuye@huawei.com