Immutable Metadata Annotation

draft-ma-netmod-immutable-flag-00

Qiufang Ma (Huawei) Presenter
Qin Wu (Huawei)
Hongwei Li (HPE)
Motivation and Goal

• Motivation
  • This idea is derived from “system-defined configuration” work
  • Some system configurations are generated to be non-modifiable to clients, while others are not
  • Allowing some configurations modifiable while others not is inconsistent and introduces ambiguity

• Goal
  • A standard mechanism to see what system configuration is read-only to clients
Solution Overview

• A metadata annotation [RFC7952] called “immutable” is defined to indicate the immutability of a data node.
  
  • The “immutable” concept can be used without being restricted to system config.
  
  • It’s used to annotate instance of YANG data nodes rather than schema nodes.
  
  • After it is created, any data node annotated with immutable=“true” is read-only to clients.

  • However, the following operations should be allowed:
    
    • Create an immutable data node with a same value initially set by the system if it doesn’t exist in the datastore;
    
    • Delete the parent node of an immutable data node unless the parent node is also annotated with immutable=“true”.
Examples

The client is not allowed to modify the interface type for interface "eth0", but the following operation should be allowed:

<predefined NAC rules in <system>
Open Issues

• Backward-compatibility: What if legacy clients receive some annotations they don’t understand?
  • Option 1: Annotations always return, but the client Ignore unknown annotations silently
  • Option 2: Define a parameter in the operation request to indicate including an “immutable” annotation in the response

• How would the client know if “immutable” is applied to the whole list, the list entries, or both? same applies to the leaf-list.

• When should the server reject modifications to immutable data node?
  • The current draft says the error reporting is performed at various different time according to the selected target ds:
    • If the target ds is <running> or <startup>, it should be in an <edit-config>/<edit-data> operation time
    • If the target ds is <candidate>, it’s delayed until a <commit> or <validate> operation takes place.

• Should we allow the client to delete an “immutable” system instantiated node in <running>?
  • There is no way to actually delete system config in <system>
    • We already define that deleteable system config must be defined in <factory-default>
    • Non-deleteable system configuration must be defined in <system>
Comments, Questions, Concerns?