Ephemeral Dynamic Datastore

An Author’s discussion

Susan Hares
1. Name : ephemeral
2. YANG modules : all (default)
3. YANG statements : config false + ephemeral true
4. How applied : automatic
5. Protocols : NC/RC (default)
6. YANG Module : (see below)

module example-ds-ephemeral {
    yang-version 1.1;
    namespace "urn:example:ds-ephemeral";
    prefix eph;

    import ietf-datastores {
        prefix ds;
    }
    import ietf-origin {
        prefix or;
    }
}
// add datastore identity
identity ds-ephemeral {
    base ds:datastore;
    description
        "The 'ephemeral' datastore.";
}

// add origin identity
identity or-ephemeral {
    base or:dynamic;
    description
        "Denotes data from the ephemeral dynamic datastore.";
}

// define ephemeral extension
extension ephemeral {
    argument "value";
    description
        "This extension is mixed into config false YANG nodes to
         indicate that they are writable nodes in the 'ephemeral'
         datastore. This statement takes a single argument
         representing a boolean having the values 'true' and
         'false'. The default value is 'false'.";
}
RESTCCCONF + Ephemeral data

• Entity tag definition for edit-collision
• If I2RS creates a dynamic datastore draft, should it discuss how section:
  – "with defaults" – How RFC8040 section 3.5.4 is adapted for dynamic work,
  – "with validation" – How validation is normally handled,
  – “how client edit collisions are handled – RFC 8040 3.4 or
Ephemeral Dynamic Database

3. Ephemeral dynamic datastores are excluded from RFC8040
   – If I2RS creates an dynamic ephemeral datastore draft, then it should discuss the following:
   – “with defaults” setting – that section 3.5.4 to not keep data after a reboot, but to provide some default setting upon being downloaded.
   – “entity tag” (RFC 3.5.2) – adaptation to provided client and priority,
   – Restriction of ephemeral dynamic datastores to exclude the “rollback
Notes page