**Simulation for next gen server management**

1. **Background**

In Intersight the next generation of Cisco servers are to be managed with Redfish (<https://www.dmtf.org/standards/redfish>) interfaces. Each server managed directly by Intersight has a Redfish server running on the BMC (Board Management Controller) which exposes set of REST APIs for Device Inventory, configuration and monitoring of the server’s hardware. Today, to test the Intersight functionality, a real hardware running RedFish server is used. This adds unnecessary cost in terms of hardware requirement as well as artificially creates limitations, where certain hardware configurations and/or certain negative conditions are not tested since real hardware does not demonstrate certain behavior. So the management code relies on the chance for certain condition to occur or failures happen when code is deployed without testing those conditions. This also brings challenges for automation testing as hardware is not always consistent in its behavior and it creates false positives, reducing the reliability of automated test cases.

1. **Goal**

To be able to create one or more simulated servers based on inventory from a physical server with specific hardware configuration that intersight device connector can connect to and get the same interface experience as a real physical server.

1. **Functional requirements**
   1. **Mockup creator**: Implement a mock up creator that pulls the inventory from a real physical server and dumps it into file system. The mock up creator shall support creating N (where N>10000) number of servers with similar configurations where unique identifiers such as Serial numbers are changed to keep them unique. The tool shall provide a User interface to specify the unique identifier and their type, and the number of mock ups needed and implement back end that creates the specified number of mockups with the updated unique identifiers. The below is an example of such a tool:

<https://github.com/DMTF/Redfish-Mockup-Creator>

* 1. **Redfish Interface emulator**: Implement a HTTP server, which can read the inventory/configuration/monitoring data from the file system and act as a redfish server. This emulator shall support, all the Cisco UCS supported Redfish APIs and corresponding HTTP functions (PUT/PATCH/POST/GET/DELETE etc.) in the same manner that a real redfish server does. The emulator shall fully support the configuration and event mechanism supported by Cisco BMC Redfish server. It shall monitor the file system and notify changes as events when they occur. Provide a tool to create N (where N>10000) Redfish servers with unique IP/port where an intersight DC can connect to it. Below is an example of such a tool: <https://github.com/DMTF/Redfish-Interface-Emulator>
  2. **Redfish Event Generator:** Implement an User interface that allows user to specify the events and the hardware change that may occur in a real physical server. Example of such events or hardware change are, a DIMM or hard drive being removed or replaced, or some hardware failure, or certain parameters crossing threshold etc. Implement a REST interface to generate these events and hardware changes, and allow for a lag in the timing on the events. POST complete should be generated 5 min from power on for ex. A REST interface would help integration with automated testing.

Reference: Cisco Redfish Schema specification <TBD>

Pip install -r requirement.txt

# a.wait(timeout=5) # block the current process, making it waiting for process a