Workshop on SNIA Swordfish Tools

Nidhi Malhotra
Microchip
Learning Objectives

- SNIA Swordfish tools
  - Swordfish API Emulator
  - Swordfish Basic Web Client
  - Swordfish Sample Dashboard Datadog Integration
  - Swordfish Sample Dashboard Power BI Integration
What is Swordfish™

- Provides a unified approach for the management of storage and servers in hyperscale and cloud infrastructure environments
- SNIA Swordfish is an extension of the DMTF Redfish specification
- Swordfish was developed by SNIA SSM TWG
Swordfish API Emulator

- Emulates a Swordfish system with storage services
- Responds to Create, Read, Update and Delete operations
- Extends the DMTF Redfish Interface Emulator
- Adds code for Swordfish resources
Installation

- Emulator python environment
- About the emulator
- Setting up the emulator
Emulator python environment

- Python 3.6 or above
- Virtualenv recommended but not mandatory
- Python packages required:
  - Flask, flask_restful, flask_httpauth, requests, aniso8601
  - markupsafe, pytz, itsdangerous, StringGenerator, urllib3
About the emulator

- The Redfish Interface Emulator README.md file tells about setting
  - Emulator.py flags and emulator-config.json
- api_emulator\resource_manager.py file establishes which emulator resources are static and which emulator resources are dynamic
  - Static resources are read-only and cannot be changed via the emulator API. These are taken from Swordfish mockup
  - Dynamic resources can be modified by CRUD operations using tools like Postman, (https://www.getpostman.com)
Setting up the Emulator

- Create a folder for the emulator
- Copy the Redfish Interface Emulator files into the emulator folder.
- Install the Python packages required by the emulator.
- Copy the Swordfish API Emulator files into the emulator folder, and allow some of the Redfish Interface Emulator files to be overwritten.
- Run with python emulator.py
Running the Swordfish API Emulator

C:\SwordfishTools\emulator\Swordfish>python emulator.py
['Redfish']
* Redfish endpoint at localhost:5000
  {'rb': '/redfish/v1/', 'sys_id': 'System-1'}
  {'rb': '/redfish/v1/', 'sys_id': 'System-2'}
  {'rb': '/redfish/v1/', 'sys_id': 'System-3'}
  {'rb': '/redfish/v1/', 'sys_id': 'System-4'}
  {'rb': '/redfish/v1/', 'sys_id': 'System-5'}
  {'rb': '/redfish/v1/', 'sys_id': 'System-6'}
  {'rb': '/redfish/v1/', 'sys_id': 'System-7'}
* Running in Redfish mode
* Serving Flask app "g" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
INFO:werkzeug: * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
Demo

- Emulator files and browser output
Swordfish Basic Web Client
Swordfish Basic Web Client

- Web client that can connect to multiple Redfish and/or Swordfish services simultaneously
- Presents the entire Swordfish hierarchy in a browser web frame
- Provides basic capabilities for viewing resources and updating properties that are writable
No Services are available
Demo

- Storage services, storage pools and corresponding storage volumes
Swordfish Datadog
Sample Dashboard Integration

- Connects to a Swordfish service and provides an integration to the Datadog User Interface
- Displays storage system capacity information and the available storage capacity thresholds
Sample Dashboard functionality

- Swordfish Dashboard views: Capacity data and threshold values for different collections like Volumes, Storage pools and filesystems.
- Data Collection: Using different custom metrics, Datadog will collect all the required data and visualize it in to a Graph or Gauge and later can configure to throw alerts and report events
Steps for Datadog Integration

- Run Emulator/server in local or host machines or vm.
- Install datadog agent everywhere – vm, server, instances
- Configure datadog
- Submit custom application metrics by writing code
- Open https://www.datadoghq.com in browser or user agent.
- Register and login to https://www.datadoghq.com
- Create dashboards and show the required data in graphs
Submit custom application metrics

- After all the installation of Datadog Agent. We need to go through /etc/dd-agent folder
- We have two Directories, one is `checks.d` and `conf.d`
- `conf.d` contains `.yaml` files & `checks.d` contains `python files`
- The agent checks and some logic are written in python and they rely on yaml file configuration.
- The name of the check file and the config file must match
Time Board

- User selects different type of graphs where just drag and drop time series widget onto dashboard
- If users want to show more flexibility with placement of graphs, can go for screenboards.
Graph Editor

1. Select your visualization
   - Timeseries
   - Heat Map
   - Distribution
   - Top List

2. Graph your data
   - a. Metric: SP_Threshold_Low, from (everywhere), avg by (everything)
   - b. Metric: SP_Threshold_Medium, from (everywhere), avg by (everything)
   - c. Metric: SP_Threshold_High, from (everywhere), avg by (everything)
   - Formula: e.g. 2 * a
   - Display: Lines, Color: Warm, Style: Solid, Stroke: Thin

3. Set display preferences
   - Show: The Past 3 Months
   - Show legend on graph, Show up to 16 entries

4. Widget title
   - Show a title: StoragePools-3- CapacitySources & Threshold
To create a ScreenBoard, just create a new dashboard and choose Screenboard.

User can control period shown on per graph basis.
Swordfish Power BI
Sample Dashboard Integration

- Connects to a Swordfish service and provides an integration to the Power BI User Interface
- Displays storage system capacity information and the available storage capacity thresholds
Access to Swordfish server

Access Web content

http://10.41.80.109:8080/redfish/v1/StorageServices/

⚠ The password won’t be encrypted when sent.

Use your Windows credentials to access this Web content.

- Use my current credentials
- Use alternate credentials

User name: Administrator

Password: ********

Select which level to apply these settings to:

http://10.41.80.109:8080/redfish/v1

Connect  Cancel
Demo

- Show bar chart creation for allocated volume and consumed bytes
<table>
<thead>
<tr>
<th>Swordfish API Emulator</th>
<th>Swordfish Basic Web Client</th>
<th>Datadog</th>
<th>Power BI</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Datacenter simulation</td>
<td>- Ease of data visualization</td>
<td>- Data monitoring</td>
<td>- Data monitoring</td>
</tr>
<tr>
<td>- Test vehicle</td>
<td>- Scripting possibility</td>
<td>- Data analytics</td>
<td>- Data analytics</td>
</tr>
<tr>
<td>- Development tool</td>
<td>- Development tool</td>
<td>- Alerts</td>
<td>- Business intelligence</td>
</tr>
<tr>
<td>- Feeder for analytics and monitoring tools</td>
<td></td>
<td>- Events</td>
<td>- Feeder for ML applications</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Graphical representation</td>
<td>- Ease of use as based on query</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Time based monitoring</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Integration with major cloud business applications</td>
<td></td>
</tr>
</tbody>
</table>
Useful links

- www.snia.org/swordfish
- http://swordfishmockups.com/
- https://github.com/SNIA
- https://github.com/SNIA/SSM
- https://github.com/SNIA/Swordfish-API-Emulator
- https://github.com/SNIA/Swordfish-basic-web-client