SDC 20

BY Developers FOR Developers

Storage Developer Conference
September 22-23, 2020

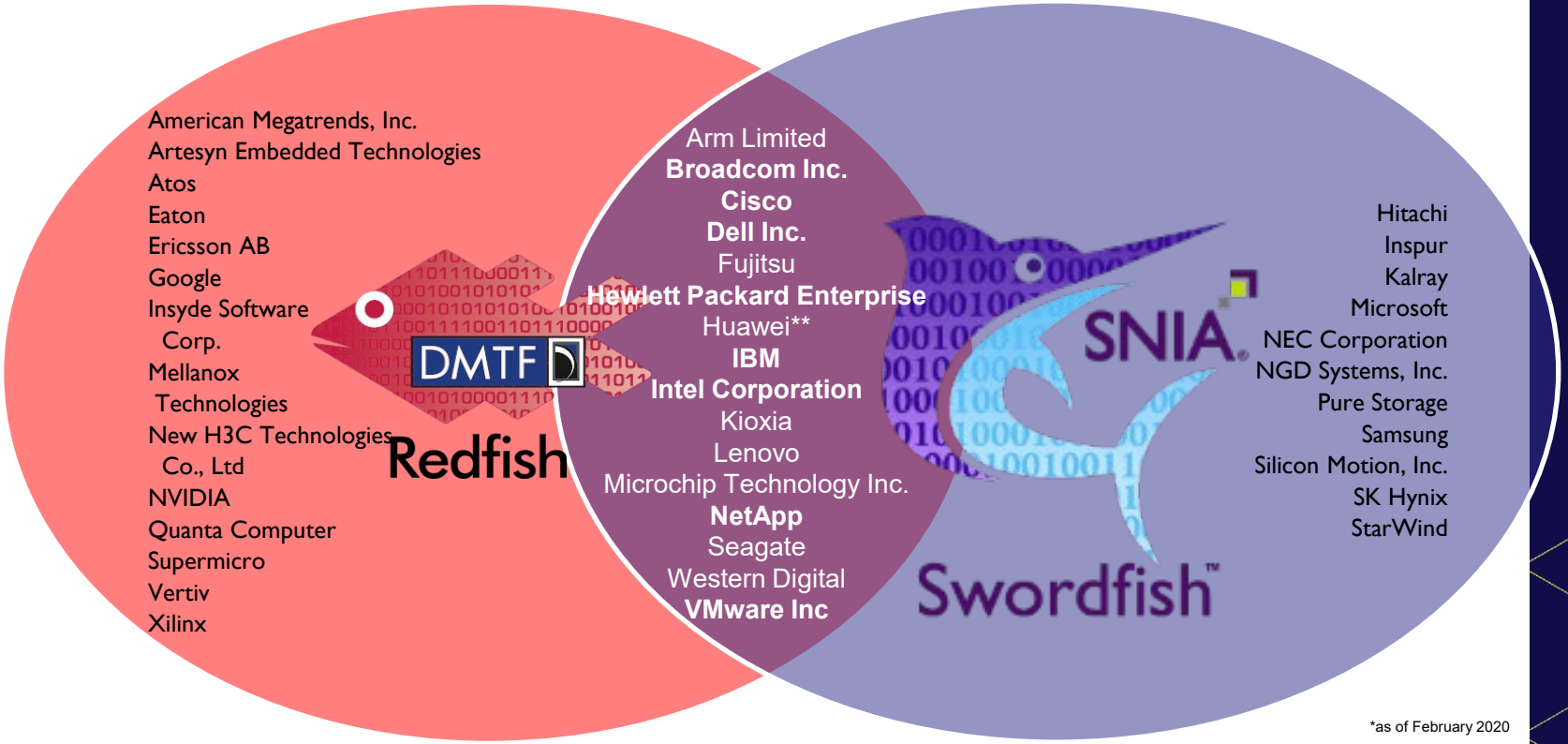
Accelerating Swordfish Implementations using OpenSource Tools

Chris Lionetti
HPE

The SNIA Swordfish™ Approach

- Develop the management model
  - point-of-view of what a client needs to accomplish
  - provide information that the client needs
- Cover block, file, and object storage
- Traditional storage domain coverage & converged environments
  - covering servers, storage and fabric together
- Implement the Swordfish API as an ***extension*** of the Redfish API
  - Build using DMTF's Redfish technologies
  - RESTful interface over HTTPS in JSON format based on OData v4

Who is Developing Redfish and Swordfish*?



*as of February 2020

** Membership suspended

Which Tools are right for you!

- Swordfish PowerShell Toolkit
- Swordfish to RestAPI Map
- Swordfish PowerShell Provider Framework
- Swordfish Emulator
- Swordfish Mockup website

Additional Projects

- Swordfish API Emulator
- Swordfish Basic Web Client
- Swordfish DataDog Sample Dashboard Integration
- Swordfish PowerBI Sample Integration
- Swordfish GOLang Library (Gofish)

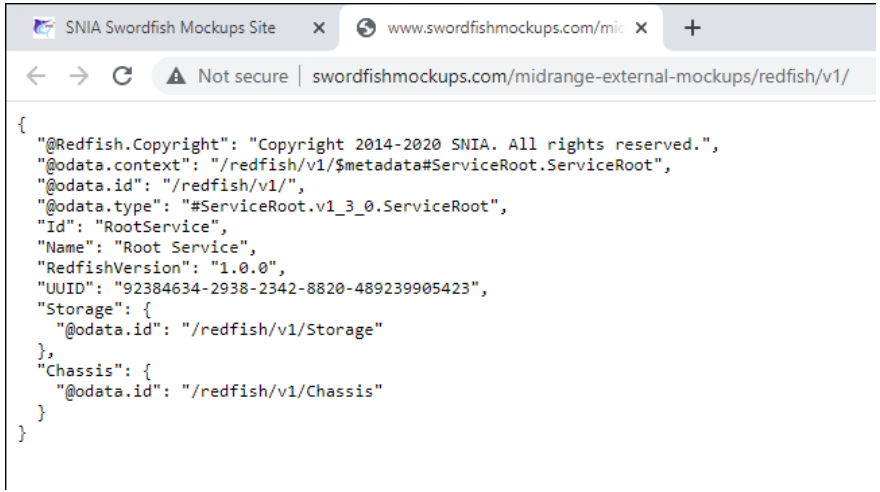
See Example Swordfish Configurations

- Technical Work Group (TWG) works with “mockups” (snapshots of a state in time) of different types of systems
- Published at http://swordfishmockups.com

Note: Mockups are representations of implementations, not normative

Overview of Swordfish Hierarchy

- Explore the Swordfish data model to see potential / typical implementation
- Navigate the model to learn about, and see, various resources
- SNIA mockups show examples of block storage systems
  - Midrange: A small external array
  - Midrange with Replication: Above model with remote replication
  - NVMe JBOF & Fabric Attached NVMe
  - Simple SSD with NVMe-oF Attach




```
{
  "@Redfish.Copyright": "Copyright 2014-2020 SNIA. All rights reserved.",
  "@odata.context": "/redfish/v1/$metadata#ServiceRoot.ServiceRoot",
  "@odata.id": "/redfish/v1/",
  "@odata.type": "#ServiceRoot.v1_3_0.ServiceRoot",
  "Id": "RootService",
  "Name": "Root Service",
  "RedfishVersion": "1.0.0",
  "UUID": "92384634-2938-2342-8820-489239905423",
  "Storage": {
    "@odata.id": "/redfish/v1/Storage"
  },
  "Chassis": {
    "@odata.id": "/redfish/v1/Chassis"
  }
}
```

What’s in a Storage System?

- Volumes
- Storage Pools
- Storage Groups
- Consistency Groups
- …
- Pointers to related resources (system, chassis, endpoints..)

Swordfish Basic Web Client Screen (StorageServices)

Not secure | 192.168.1.146:3000/#/home

SAE180828 > StorageServices

Swordfish Service Add Remove

- SAE180828

Explore The Resources

- Chassis
- Managers
- TaskService
- SessionService
- StorageServices
- StorageSystems
- AccountService
- EventService
- Registries
- Systems
- CompositionService

StorageServices Add Remove

- 1
- 2
- AFF-1

Properties

Name : Storage Service Collection

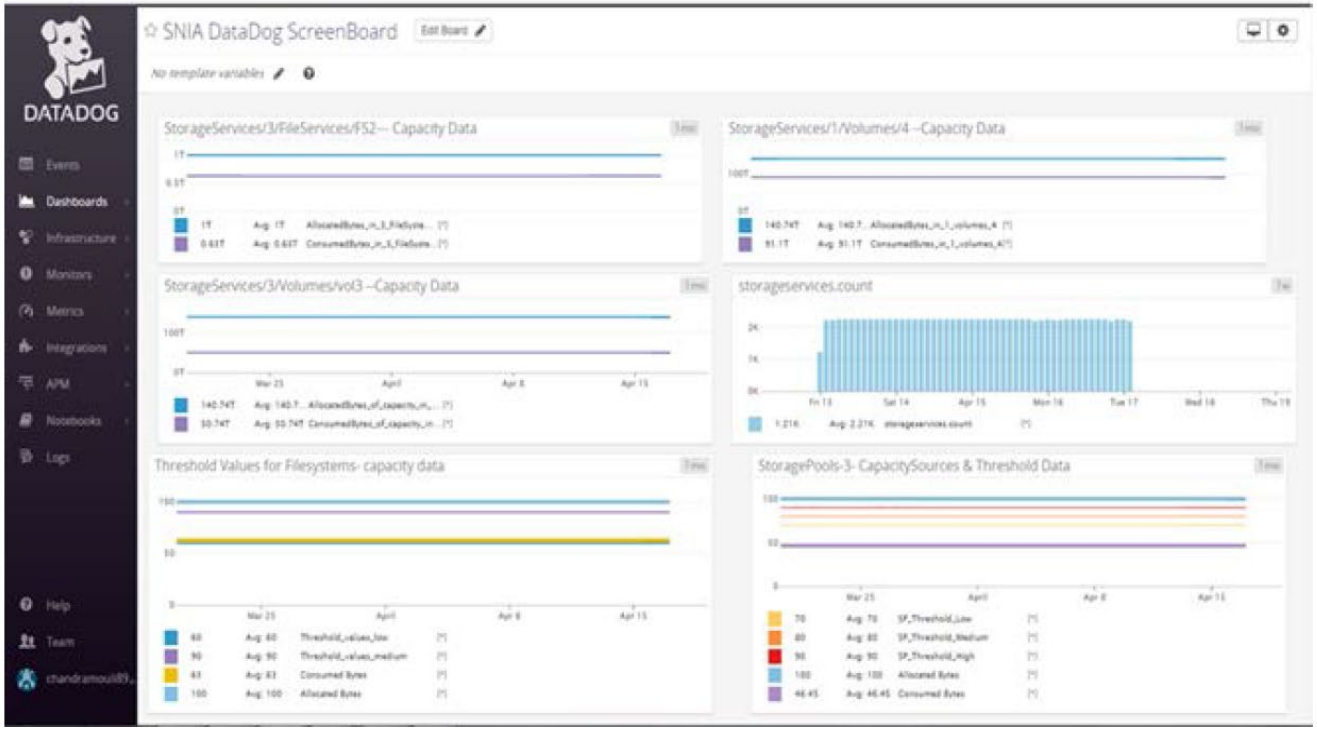
ODATA

LINKS

Swordfish Datadog Sample Dashboard Integration

- Basic dashboard for the Datadog monitoring service
- 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
- Can be a starting point for a customized Datadog plugin
- Link: https://github.com/SNIA/Swordfish-datadog-sample-dashboard-integration
- Includes installation, user, and developer documentation

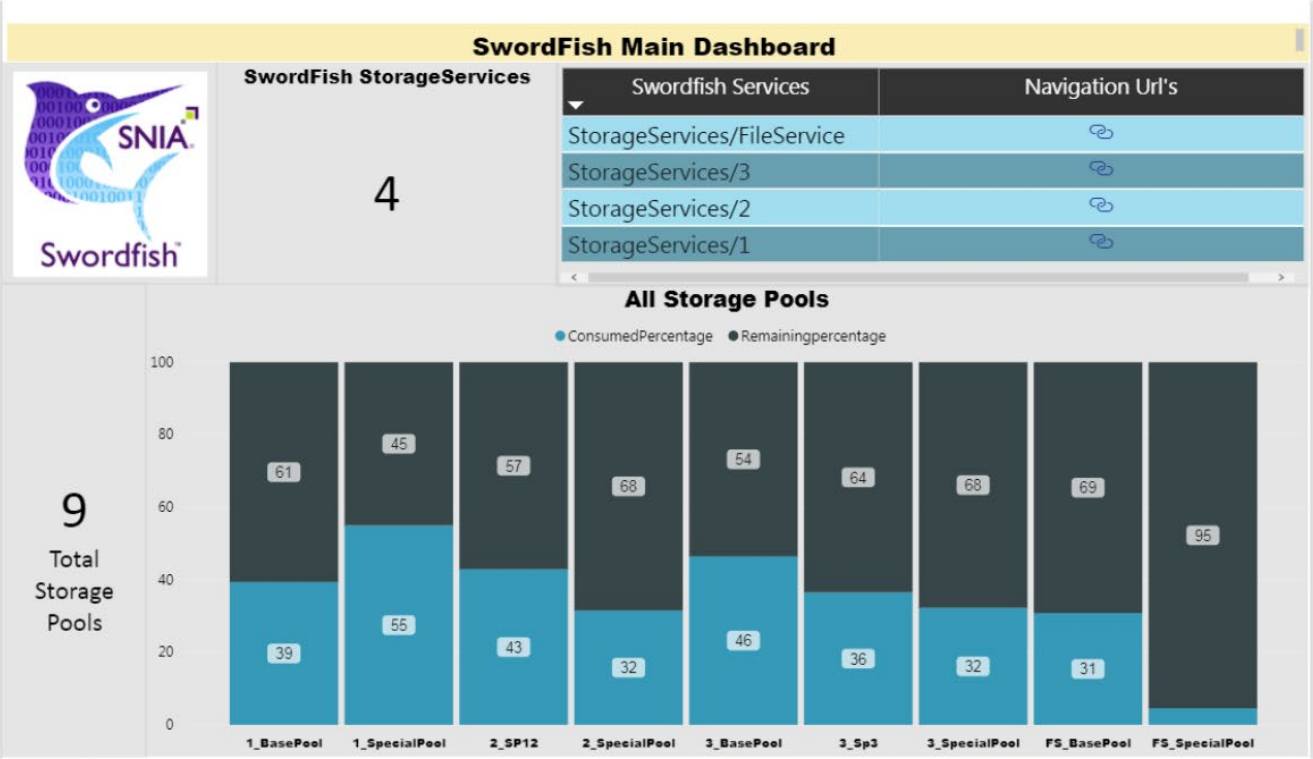
Swordfish Datadog Sample Dashboard Output



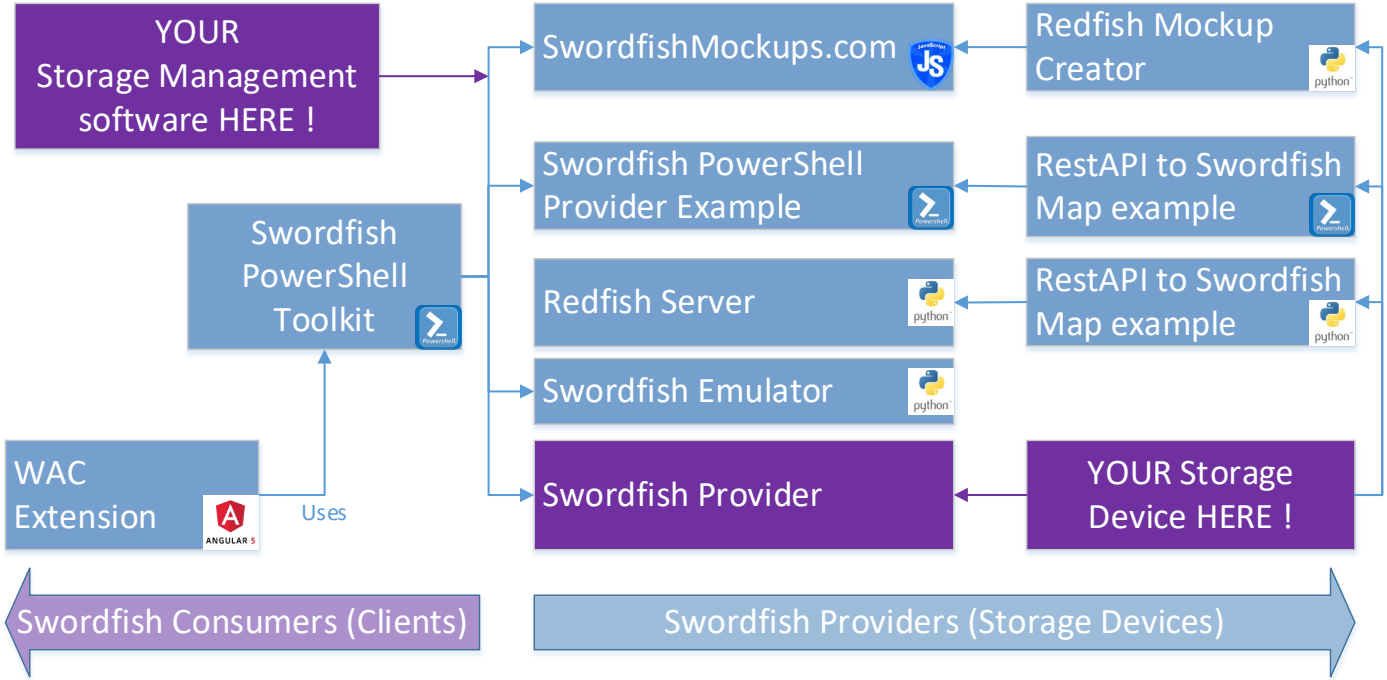
Swordfish Power BI Sample Dashboard Integration

- Basic dashboard for the Power BI monitoring system
- 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
- Can be a starting point for a customized Power BI plugin
- Link: https://github.com/SNIA/Swordfish-powerBI-sample-dashboard-integration
- Includes installation, user, and developer documentation

Swordfish Power BI Sample Dashboard (Main)



Which Tools are right for you!



SNIA SWORDFISH™ POWERSHELL TOOLKIT

WHAT IS THE POWERSHELL TOOLKIT?

- Open source project between HPE and Pure Storage
  - https://github.com/SNIA/Swordfish-Powershell-Toolkit
- Supported on Windows Server, Linux and macOS
  - Can query a Swordfish Target, A simulator, or even SwordFishMockup.com
- PowerShell wrapper for REST API calls to Redfish and Swordfish

Administrator: Windows PowerShell

```
PS C:\Users\Administrator\Desktop\Swordfish-Powershell-Toolkit> import-module .\SNIASwordFish.psm1
PS C:\Users\Administrator\Desktop\Swordfish-Powershell-Toolkit> Connect-SwordFishTarget -Target 'localhost' -Port 5000
Base URI = http://localhost:5000/redfish/v1/

@odata.context : /redfish/v1/$metadata#ServiceRoot
@odata.type    : #ServiceRoot.1.0.0.ServiceRoot
@odata.id      : /redfish/v1/
Id             : RootService
Name           : Root Service
ServiceVersion : 1.0.0
UUID
Links
PS C:\User
```

Administrator: Windows PowerShell

```
PS C:\Users\Administrator\Desktop\Swordfish-Powershell-Toolkit> import-module .\SNIASwordFish.psm1
PS C:\Users\Administrator\Desktop\Swordfish-Powershell-Toolkit> Connect-SwordFishMockup

@Redfish.Copyright : Copyright 2014-2019 Distributed Management Task Force, Inc. (DMTF). All rights reserved.
@odata.context     : /redfish/v1/$metadata#ServiceRoot.ServiceRoot
@odata.id          : /redfish/v1/
@odata.type        : #ServiceRoot.v1_3_0.ServiceRoot
```

Powe

- Everything is retu
  - Cast to Variab
  - Can filter by p
  - Can dig deep
  - And you can

```
PS C:\> $MyVols[4] | convertto-json
{
    "@Redfish.Copyright":  "Copyright 2014-2019 SNIA. All rights reserved.",
    "@odata.context":  "/redfish/v1/$metadata#Volume.Volume",
    "@odata.id":  "/redfish/v1/StorageServices/1/Volumes/5",
    "@odata.type":  "#Volume.v1_4_0.Volume",
    "Name":  "Volume 5",
    "Id":  "5",
    "Description":  "Volume 5.",
    "Identifiers":  [
                        {
                            "DurableNameFormat":  "NAA",
                            "DurableName":  "65456765456761001244076100123487"
                        }
                    ],
    "Manufacturer":  "SuperDuperSSD",
    "Model":  "Drive Model string",
    "Status":  {
                    "State":  "Enabled",
                    "Health":  "OK"
                },
    "AccessCapabilities":  [
                               "Read",
                               "Write",
                               "Append",
                               "Streaming"
                           ],
    "BlockSizeBytes":  512,
    "CapacitySources":  [
                            {
                                "@odata.id":  "/redfish/v1/StorageServices/1/Volumes/5#/CapacitySources/0",
                                "MemberId":  "0",
                                "ProvidedCapacity":  "@{ConsumedBytes=0; AllocatedBytes=10737418240; GuaranteedBytes=536870912; P
                                "ProvidingPools":  ""
                            }
                        ],
    "Capacity":  {
                     "Data":  {
                                  "ConsumedBytes":  0,
                                  "AllocatedBytes":  10737418240,
                                  "GuaranteedBytes":  536870912,
                                  "ProvisionedBytes":  1099511627776
```

PowerShell Command Help

- Get a list of valid commands
- Get Help on a specific command
  - Option to show examples
  - Option to show All
- Verbose option to see raw transactions

```
Get-Command -Module SNIASwordFish
```

```
get-help Get-SwordFishVolume
get-help Get-SwordFishVolume -Examples
get-help Get-SwordFishVolume -Full
```

```
Get-SwordFishVolume -Verbose $true
```

Administrator: Windows PowerShell

```
PS C:\> get-help Get-SwordFishVolume -Full

NAME
    Get-SwordFishVolume
SYNOPSIS
    Retrieve The list of valid Volumes from the SwordFish Target.
SYNTAX
    Get-SwordFishVolume [[-StorageServiceID] <String>] [[-VolumeId] <String>] [<CommonParameters>]
DESCRIPTION
    This command will either return the a complete collection of Volumes that exist across all of
    the Storage Services, unless a  specific Storage Service ID is used to limit it, or a specific
    Volume ID is directly requested.
PARAMETERS
    -StorageServiceID <String>
```

PowerShell Toolkit Work Items

The PowerShell Toolkit commands;

- Get-SwordFishChassis (+ Power, +Thermal)
- Get-SwordFishDrive
- Get-SwordFishEndpoint
- Get-SwordFishEndpointGroup
- Get-SwordFishStoragePool
- Get-SwordFishStorageService
- Get-SwordFishVolume
- Get-SwordFishClassOfService
- Connect-SwordFishTarget
- Connect-SwordfishMockup

Command sets that need to be written;

(in order of priority)

- New/Set/Remove-SwordFishEndpoint
- New/Set/Remove-SwordFishEndpointGroup
- New/Set/Remove-SwordFishStoragePool
- New/Set/Remove-SwordFishStorageGroup
- New/Set/Remove-SwordFishConsistencyGroup
- New/Set/Remove-SwordFishVolume
- Set-SwordFishStorageService
- Set-SwordFishChassis
- Get/New/Set/Remove-*LoS
- New/Set/Remove-SwordFishClassOfService

- Common Nomenclature
  - RestAPI vs PowerShell. Create = New, Read = Get, Update = Set, Delete = Remove
- All Commands must have inline help before being checked into the build
- All Commands must work against BOTH the Swordfish Targets (directly) and SwordFishMockups.com
- All Commands are open source, no compiled code or external DLL dependencies

RestAPI to SWORDFISH™ Mapping

- RestAPI Docume
- PowerShell Toolk
- Basic PowerShell

Steps

1. Retrieve the Volu
2. Hold it side-by-s
3. Look for Matche similarities

   i.e. You may s Bytes. You wo relevant sworc
4. Go through the r using your result

Administrator: Windows PowerShell

```
PS C:\> Get-NSVolume -name SCSCOM2019 | convertto-json
{
    "agent_type":  "none",
    "app_category":  "Other",
    "app_uuid":  "",
    "avg_stats_last_5mins":  {
                                "combined_iops":  0,
                                "combined_latency":  66,
                                "combined_throughput":  692,
                                "read_iops":  0,
                                "read_latency":  0,
                                "read_throughput":  0,
                                "write_iops":  0,
                                "write_latency":  66,
                                "write_throughput":  692
                              },
    "base_snap_id":  "",
    "base_snap_name":  "",
    "block_size":  4096,
    "cache_needed_for_pin":  107374182400,
    "cache_pinned":  false,
    "cache_policy":  "normal",
    "caching_enabled":  true,
    "cksum_last_verified":  0,
    "clone":  false,
    "content_repl_errors_found":  false,
    "creation_time":  1559321039,
    "dedupe_enabled":  false,
    "description":  "Data Volume for SCOM 2019",
    "multi_initiator":  false,
    "name":  "SCSCOM2019",
    "needs_content_repl":  false,
    "num_connections":  2,
    "num_fc_connections":  0,
    "num_iscsi_connections":  2,
    "num_snaps":  69,
    "offline_reason":  null,
    "online":  true,
    "online_snaps":  null,
    "owned_by_group":  "Firefly",
    "owned_by_group_id":  "002b4bd8361b856bbc000000000000000000000001",
    "parent_vol_id":  "",
    "parent_vol_name":  "",
    "perfpolicy_id":  "032b4bd8361b856bbc000000000000000000000010",
    "perfpolicy_name":  "Other Workloads",
    "pinned_cache_size":  0,
    "pool_id":  "0a2b4bd8361b856bbc000000000000000000000001",
    "pool_name":  "default",
    "previously_deduped":  false,
```

JSON | Raw Data | Headers

swordfishmockups.com/redfish/v1/StorageServices/ISC/Volumes/1/

```
{
  "@Redfish.Copyright": "Copyright 2015-2019 SNIA. All rights reserved.",
  "@odata.context": "/redfish/v1/$metadata#Volume.Volume",
  "@odata.id": "/redfish/v1/StorageServices/ISC/Volumes/1",
  "@odata.type": "#Volume.v1_2_1.Volume",
  "Id": "1",
  "Name": "Logical Disk 1",
  "Identifiers": [{
    "DurableNameFormat": "UUID",
    "DurableName": "123e4567-a12b-12a3-a123-123456789000"
  }],
  "Manufacturer": "BEST ChipCo",
  "Status": {
    "State": "Enabled",
    "Health": "OK"
  },
  "BlockSizeBytes": 512,
  "VolumeType": "Mirrored",
  "Capacity": {
    "Data": {
      "ConsumedBytes": 1099511627776,
      "AllocatedBytes": 1198027440128
    }
  },
  "CapacitySources": [{
    "@odata.id": "/redfish/v1/StorageServices/ISC/Volumes/1#/CapacitySources/0",
    "MemberId": "0",
    "ProvidedCapacity": {
      "AllocatedBytes": 1198027440128,
      "ConsumedBytes": 1198027440128
    },
    "ProvidingPools": [{
      "@odata.id": "/redfish/v1/StorageServices/ISC/StoragePools/SASPool"
    }]
  }]
```

Some values are hardcoded per spec

Other values are partially hardcoded with known values added

Create a File Structure to match Swordfish

SEE HTTPS://GITHUB.COM/CHRIS-LIONETTI/SWORDFISHMOCKUP

- Using PowerShell you can create a function for each thing you wish to express in SwordFish.
- Make a master script that runs you function against all things in your device.
- Create PowerShell Objects that can be converted to JSON as saved as Index.json files.
- In example to right, Variables all start with ‘$’ and constants are shown in brown.

```
$VolObj =@{'@Redfish.Copyright'      =   $RedfishCopyright;
           '@odata.context'          =   '/redfish/v1/$metadata#Volumes/'+$NimbleSerial+'/Volumes/'+$Snapshot.name;
           '@odata.id'               =   '/redfish/v1/$metadata#Volumes/'+$NimbleSerial+'/Volumes/'+$Snapshot.name;
           '@odata.type'             =   '#Volumes_1_4_0.Volume';
           Id                        =   $Snapshot.id;
           Name                      =   $Snapshot.name;
           Description               =   $Snapshot.description;
           Capacity                  =   @{  AllocatedBytes  =   ($Snapshot.Size * 1024)
                                          };
           Status                    =   @{  State           =   $SnapStatus_state;
                                             Health          =   $SnapStatus_health;
                                          };
           BlockSizeBytes            =   $Volume.block_size;
           MaxBlockSizeBytes         =   $Volume.block_size;
           OptimumIOSizeBytes        =   $Volume.block_size;
           Manufacturer              =   'HPENimbleStorage';
           Encrypted                 =   $Vol_Encryption;
           EncryptionTypes           =   'ControllerAssisted';
           ProvisioningPolicy        =   'thin';
           Compressed                =   'true';
           Deduplicated              =   $Volume.dedupe_enabled;
           DisplayName               =   $Volume.Full_name+'+'+$Snap.name;
           LowSpaceWarningThresholdPercents =  $Volume.warn_level;
           VolumeType                =   'Snapshot';
           VolumeUsageType           =   "Data";
           ReadCachePolicyType       =   $Vol_CachePolicy;
           WriteCacheState           =   'Enabled'
           WriteCachePolicyType      =   "ProtectedWriteBack";
           WriteCacheStateType       =   "Protected";
           WriteHoleProtectionPolicyType = "Journaling";
```

SWORDFISH™ PowerShell Provider Example

How to Serve Swordfish…It's a Cookbook!

SEE HTTPS://GITHUB.COM/CHRIS-LIONETTI/SWORDFISHMOCKUP

- Codebase Assumes that you have created a Mockup that runs against the output of that mockup.
  - The Mockup can save to hard drive andact as a static reference.
  - The Mockup can be directed to pull live information for each Swordfish request.
- Code is hidden command in the Mockup called 'Listener.ps1'

```
# Create a listener on port 5000
$listener = New-Object System.Net.HttpListener
$listener.Prefixes.Add('http://+:5000/')
$listener.Start()
write-host 'Listening ...To end this session connect to the IP Address with
# Run until you send a GET request to /end
```

```
Administrator: Windows PowerShell
PS C:\Users\chris\Desktop\SwordfishMockup> .\listener.ps1
WARNING: Ensure that Port 5000 is currently not listened to.
Successfully connected to array 192.168.1.60

WARNING: Connected to Array
Listening ...Go Swordfish Go.
```

Breaking News

- HPE MSA 2060 (Gen 6)
  - 1st Storage Array to natively support Redfish/Swordfish
  - Released Sept 8th 2020

- This deck will be refreshed over time to add more vendors as their solutions become available to the public.

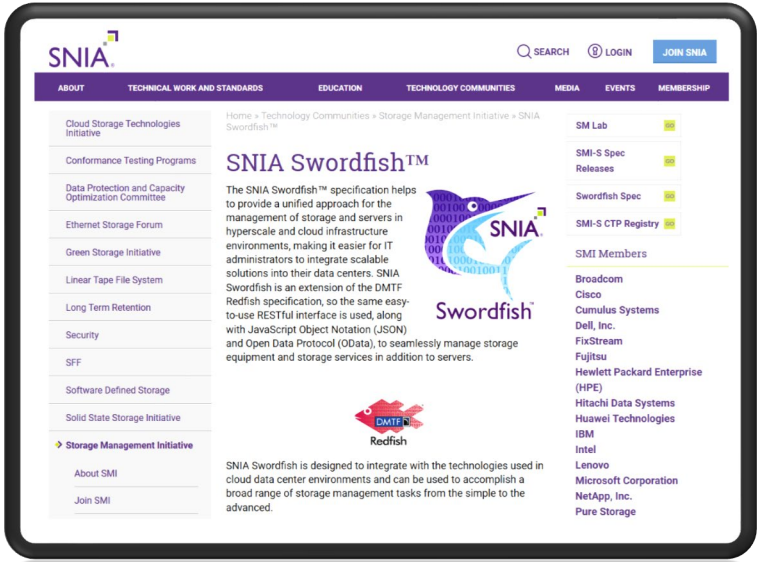
https://192.168.100.98/redfish/v1

Not secure | 192.168.100.98/redfish/v1


```
{
    "@odata.context":"/redfish/v1/$metadata#ServiceRoot.ServiceRoot",
    "@odata.id":"/redfish/v1/",
    "@odata.type":"#ServiceRoot.v1_2_0.ServiceRoot",
    "Id":"RootService",
    "Name":"Root Service",
    "RedfishVersion":"1.0.2",
    "UUID":"92384634-2938-2342-8820-489239905423",
    "Systems":    {
        "@odata.id":"/redfish/v1/ComputerSystem"
    },
    "Chassis":    {
        "@odata.id":"/redfish/v1/Chassis"
    },
    "StorageServices":    {
        "@odata.id":"/redfish/v1/StorageServices"
    },
    "Managers":    {
        "@odata.id":"/redfish/v1/Managers"
    },
    "Tasks":    {
        "@odata.id":"/redfish/v1/TaskService"
    },
    "SessionService":    {
        "@odata.id":"/redfish/v1/SessionService"
    },
    "Links":    {
        "Oem":{}
,
        "Sessions":        {
            "@odata.id":"/redfish/v1/SessionService/Sessions"
        }
    }
}
```

Swordfish Info: www.snia.org/swordfish

- Resources
  - Specifications
  - User's Guide
  - GitHub for Swordfish Tools
  - Practical Guide
  - Other Documentation
- Swordfish Mockups Site
  - ISC and HSC configurations
  - Block vs file configurations
  - Small and large configurations
- Education/Community
  - Whitepapers, Presentations
  - YouTube shorts & Webinars
- Participate
  - Join SNIA and the SSM TWG
  - Implement



Next Steps

- Develop a Swordfish Mockup of your own & submit it to the Swordfish forum;
  - Feedback on spec adherence to validate your mockup.
  - Will be posted as an additional example in the SwordfishMockups.com site.
- Join SNIA and the SSM TWG & help define the Schema;
  - Ensure the Schema is defined sufficiently to represent your desired implementation
    - WE ARE ALWAYS LOOKING FOR FEEDBACK REGARDING YOUR IMPLEMENTATION MAPING TO SWORDFISH !
  - Full NVMe Enablement: Functionality alignment across DMTF, NVMExpress/NVMe-MI and SNIA for NVMe use cases
  - Enhanced profile support for SNIA Alliance partner organizations
- Help define the future of this Swordfish Consumer.
  - SwordFish™ PowerShell Toolkit and follow-on Windows Admin Client Module.
  - notable projects; Swordfish DataDog implementation & PowerBI
  - A GoLang Client library called GoFish; An EmberJS Client
  - Looking for more integration points (what can you come up with)

Q&A

Thanks to the SSM TWG members for additional slide content

Please take a moment to rate this session.

Your feedback matters to us.