SANBlaZe
an ellisys company

White Paper

SR-IOV Device Testing with SBExpress-RM4 and RM5

Testing Multi-Function PCIe Devices
with *Certified by SANBlaze* and SBExpress - V0.3

Authors:

Vince Asbridge, CEO SANBlaze

Scott Gilmour, Software Qualification Engineer, SANBlaze

Version 0.1 - First Review Copy

Version 0.2 - Incorporated Review Feedback

Steve E. – Done
Bill A. – Done, but waiting for examples of API and CLI
Vince A. – Done
Patty B. – Done

Version 0.3 – Updates

Scott Gilmour – updated white paper and added CLI examples
Patty B – edits and formatting

Table of Contents

Table of Figures

Introduction

Modern PCIe NVMe devices introduce the capability of Multi-Function PCIe devices, whereby each physical PCIe device creates multiple "child" devices. From the standpoint of software and NVMe device management these Multi-Function devices create Virtual Functions (VFs) that appear on PCIe and look to the operating system like an independent PCIe device.

While Virtual Functions (VFs) are a convenient way to manage storage, allowing each VF to have characteristics independent of the parent device, they introduce unique challenges from a testing standpoint for the host Operating System.

Version V10.5 of SANBlaze NVMe software, SBExpress and Certified by SANBlaze test suites adds the capability of configuring, managing and testing Virtual Function devices.

This white paper will describe how to manage devices with Virtual Functions, such as Single Root I/O Virtualization (SR-IOV) and how to build test reports for these devices.

Overview of Single Root I/O Virtualization (SR-IOV)

Typically used in a Virtual environment such as VMware, OpenStack or Hyper-V, SR-IOV devices provide a virtual hardware environment well suited to the virtualized system configuration. Referencing the link below, find a description of the benefits of VFs in a Virtualized software stack.

The single root I/O virtualization (SR-IOV) interface is an extension to the PCI Express (PCIe) specification. SR-IOV allows a device, such as a network adapter, to separate access to its resources among various PCIe hardware functions. These functions consist of the following types:

- A PCIe Physical Function (PF). This function is the primary function of the device and advertises the device's SR-IOV capabilities. The PF is associated with the Hyper-V parent partition in a virtualized environment.
- One or more PCIe Virtual Functions (VFs). Each VF is associated with the device's PF. A VF shares one or more physical resources of the device, such as a memory and a network port, with the PF and other VFs on the device. Each VF is associated with a Hyper-V child partition in a virtualized environment.

Each PF and VF is assigned a unique PCI Express Requester ID (RID) that allows an I/O memory management unit (IOMMU) to differentiate between different traffic streams and apply memory and interrupt translations between the PF and VFs. This allows traffic streams to be delivered directly to the appropriate Hyper-V parent or child partition. As a result, nonprivileged data traffic flows from the PF to VF without affecting other VFs.

SR-IOV enables network traffic to bypass the software switch layer of the Hyper-V virtualization stack. Because the VF is assigned to a child partition, the network traffic flows directly between the VF and child partition. As a result, the I/O overhead in the software emulation layer is diminished and achieves network performance that is nearly the same performance as in nonvirtualized environments.

Reference: https://docs.microsoft.com/SR-IOV

Getting Started with PCIe NVMe Virtual Functions

NVMe SR-IOV devices bring to the storage stack similar benefits network SR-IOV devices bring, allowing the software stack to access each VF as if it were a physical device, with the IO flowing directly between the virtual machine and the VF device and allowing system performance on par with physical NVMe devices.

From a software and testing perspective SR-IOV devices introduce unique challenges, because Virtual Functions (VFs) appear to the system exactly as a true physical device would, but they have unique characteristics which must be considered while testing.

For example, for SR-IOV devices consider:

- At start of day, an SR-IOV capable device will need to be configured
- Although each VF will appear as independent, they have common attributes
    - Power for the parent will disable parent plus all VFs
    - The physical PCIe connection is common to parent and all VFs
    - PCIe reset (PERST) will reset the parent and all VFs
- The system BIOS will see multiple PCIe devices and will need to support SR-IOV

The remainder of this document will describe the SANBlaze implementation of SR-IOV and how to configure your system to test these complex devices.

A certain level of prior knowledge of the SANBlaze NVMe test system is assumed for the remainder of the document.

Your system must be fully licensed for NVMe and be running at least version V10.5 software.

Accessing the SANBlaze System

The SANBlaze NVMe test system is accessed via a web browser and by SSH to the Command Line Interface. Both methods are used for the remainder of this document and are described here.

Access via a Browser

Point your web browser to the IP address of the SANBlaze system as shown below:

http://192.168.1.118/home.asp

Default login information is as follows:

Username: system
Password: SANBlaze

The landing page contains basic system information, and it is assumed by this document that the system has been configured for network access and is running at least V10.5 software as shown below:

SANBlaze VirtuaLUN System Status			
Hostname	Qual-DT4-Centos-1-1	SW Version	V10.5-64-dev-C8 built on Jul 20 2022 at 16:16:28
IP Address DHCP: ☑	192.168.1.134	Kernel Release	4.9.107
Gateway	192.168.1.1	HW Address	00:13:95:3B:AD:1E
Netmask	255.255.255.0	Management Port	eth0
Date	07/26/2022	Time	09:08:10 AM
Timezone	(GMT -5:00) US & Canada Eastern Time		
NTP server Enable: ☑	time.nist.gov	DNS address	192.168.1.11

Figure 1: System Home Page

Should you be running a version prior to V10.5, contact SANBlaze sales for the latest release.

While on the landing page, be sure SBExpress NVMe is enabled, it will be used later in the document:

SBExpress NVMe Support	☑

Figure 2: Enable SBExpress NVMe Support

If SBExpress NVMe is not enabled, enable it by selecting the check box and rebooting the system from the Poweroff/Reset page from the left-hand menu.

Access via an SSH Session

Examples below will use the Command Line Interface (CLI) to the system, so also create an SSH session using your favorite client (e.g. putty) or by the built-in SSH client from the landing page as shown below.

Command Shell	Open Shell

Figure 3: Using Open Shell for CLI Access

Select the Open Shell hyperlink to establish an SSH session using the credentials:

Username: vlun
Password: SANBlaze

Once the session is established, you must gain root access to complete the examples below. Do not skip this step, and if your system administrator has changed the root password, you'll need to contact her for these credentials.

$ su

Username: root
Password: SANBlaze

#

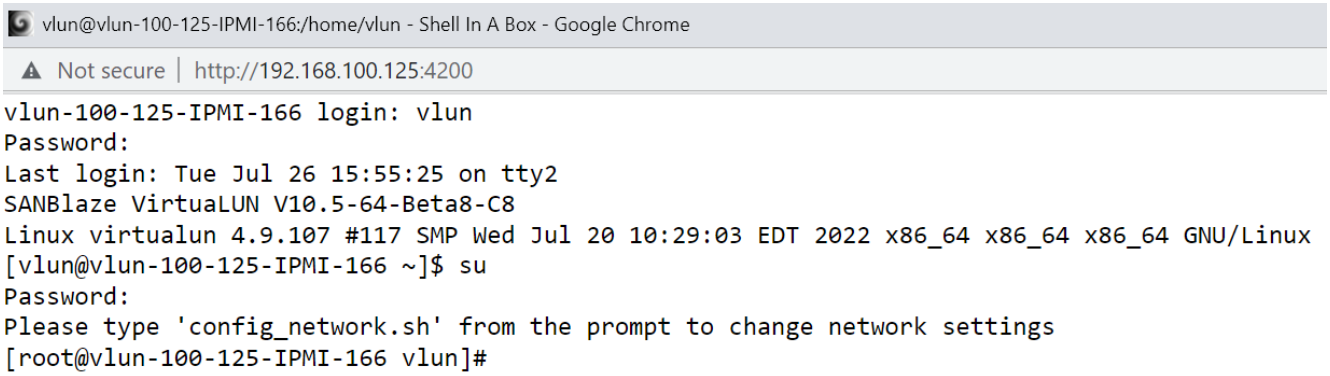


Figure 4: Logging in via Open Shell

Displaying SR-IOV Virtual Functions at the Web and CLI

Once you have established both web and CLI access to the SANBlaze system and have enabled root access at the CLI, you can use the examples below to configure and test your Multi-Function device.

From the left-hand menu, you will see by default that all NVMe devices currently seen by the system are enumerated. Also note that the SR-IOV device is not configured with any children (VFs) at this point. The SR-IOV capable drive is at target 101.

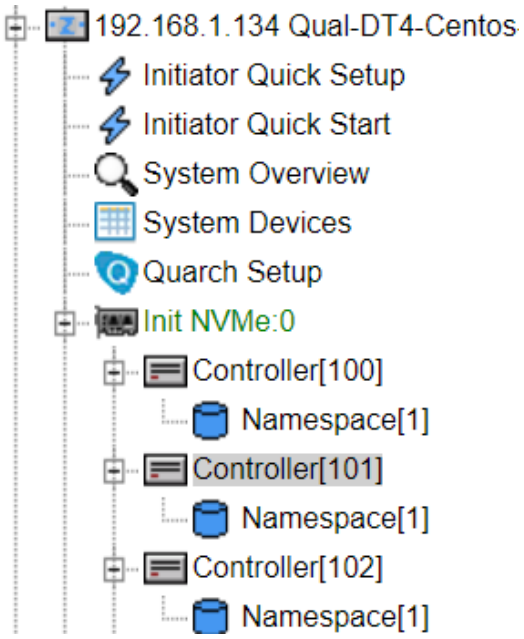


Figure 5: NVMe Devices in Left-hand menu

Selecting the "Init NVMe:0" tab in the menu will expose basic information with respect to the NVMe configuration of the system.

NVMe Controllers						
	PCI Name	PCI Info	Upstream PCI Express Port			
	0000:06:00.0	Speed 8GT/s, Width x16	SANBlaze Technology, Inc. - DT4			
Controller Name	PCI Name	PCI Info	Model Number	Serial Number	Firmware	Target
nvme5	0000:18:00.0	Speed 8GT/s, Width x4	SAMSUNG MZVLW128HEGR-000L1	S341NX0K864629	5L1QCXB7	100
nvme4	0000:16:00.0	Speed 16GT/s, Width x2	SAMSUNG MZWLJ1T9HBJR-00007	S4YNNG0R500540	EPK9AB5Q	101

Figure 6: Selecting Init NVMe:0 shows Associating Target with PCIe ID

Note the PCIe information for nvme2 16:00.0. This PCIe ID can be used to identify the SR-IOV parent at the CLI of the system as shown below.

From the CLI and enter the following command:

```
lspci | grep -i Non
0a:00.0 Non-Volatile memory controller: Intel Corporation NVMe DC SSD [3DNAND, Beta Rock Controller]
0e:00.0 Non-Volatile memory controller: Phison Electronics Corporation E16 PCIe4 NVMe Controller (rev 01)
12:00.0 Non-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller SM961/PM961
14:00.0 Non-Volatile memory controller: KIOXIA Corporation NVMe SSD Controller Cx6 (rev 01)
```

```
16:00.0 Non-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller PM173X
17:00.0 Non-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller PM173X
18:00.0 Non-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller SM961/PM961
```

Figure 7: CLI Command lspci | grep -I Non

And note our SR-IOV device at 16:00.0

At this point, there are seven NVMe devices in our system and no Virtual Functions

Using a "Nickname" for a Controller

For the purpose of example, I'll use the "Controller Nickname" function to name our SR-IOV capable controller. This "Nickname" will be associated with the controller and make the device easier to identify in the subsequent examples. Note, this step is optional and Nickname can be any string. For example the Nickname could be "Vince" to identify a controller as in use by Vince.

From the Left-hand menu, select the "Controller 101", yours may differ.

Figure 8: Controller 101 in the Left Hand Menu

Select the first tab "Controller 101 Status"

Add the Nickname "SR-IOV" (optional step) and select the "Apply" button:

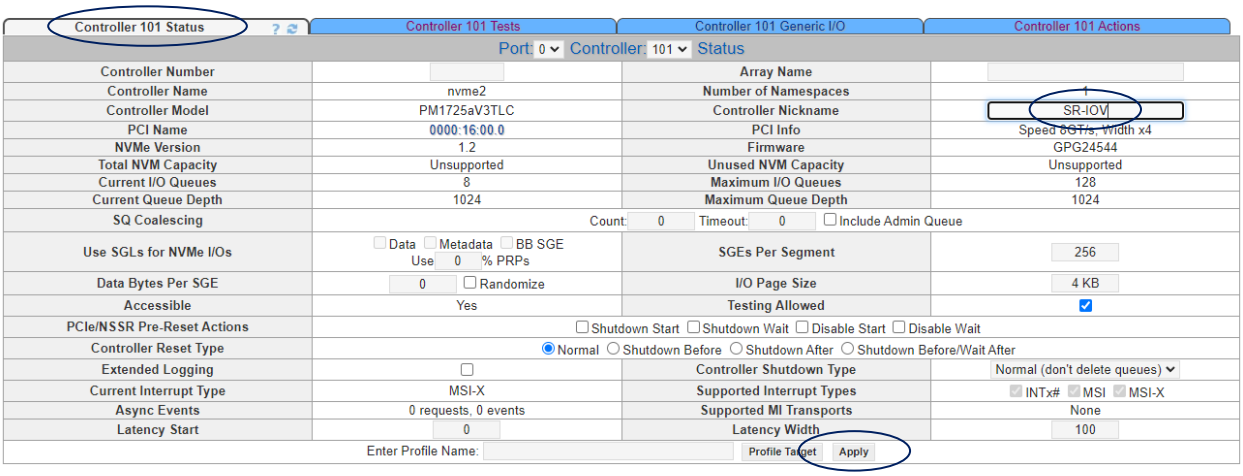


Figure 9: Applying a Nickname to a Controller

And the Nickname is associated with the Controller at 101 in the Left-hand menu and on the SBExpress page.

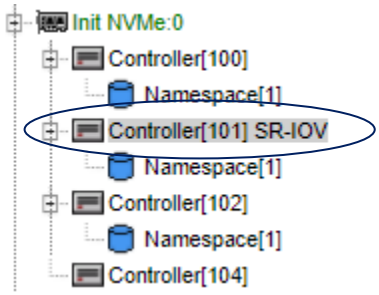


Figure 10: Nickname in Left-hand menu

From the CLI enter the following command:

Create a Controller Nickname

```
[root@Qual-DT4-Centos-1-134 ~]# sbecho Nickname=SR-IOV > /iport0/target101
[root@Qual-DT4-Centos-1-134 ~]# grep Nickname /iport0/target*
/iport0/target100:Nickname=
/iport0/target101:Nickname=SR-IOV
/iport0/target102:Nickname=
/iport0/target103:Nickname=
/iport0/target104:Nickname=
/iport0/target105:Nickname=
/iport0/target201:Nickname=
[root@Qual-DT4-Centos-1-134 ~]#
```

Figure 11: From CLI create Controller Nickname and grep to show results

Viewing your SR-IOV Device on the SBExpress Page

If the system has an SBExpress RM4, RM5, DT4 or DT5 attached and SBExpress has been enabled as shown above, your left-hand menu will have a tab for "SBExpress Manager". Select SBExpress to access a visual representation of the system and the tests associated with each controller and namespace.

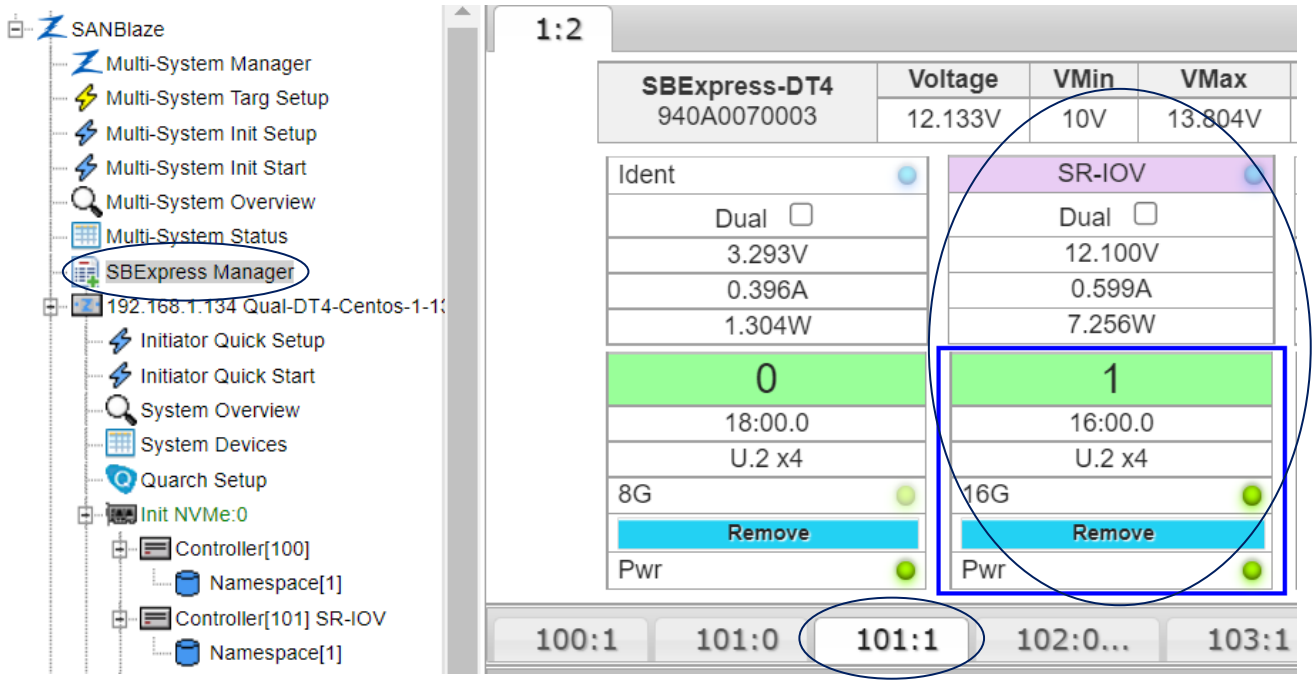


Figure 12: SR-IOV Device in SBExpress Manager

In the figure above, note that while the device at slot 1, with controller 101 is SR-IOV capable, there are no Virtual Functions enabled on the device, and therefore there is one Namespace tab associated with the device. In the example above, the tab is labeled 101:1 (controller:namespace). To get to 101:1 you must

first click on the slot number (which is circled) and that takes you to 101:0 by default, then just click on 101:1 to get to that tab.

Using sb_logger to Log System Activity

Before adding our Virtual Functions, first let's enable sb_logger at the CLI so we can monitor the devices as they arrive.

Note*:* Many users leave sb_logger running in an SSH window whenever using the SANBlaze system as a means of monitoring what's going on**.**

From the Open Shell prompt or your SSH client's CLI start sb_logger. Note, this program will stay running, you will need to use ^C (Ctrl C) to exit:

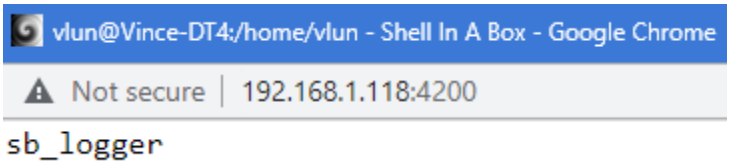


Figure 13: Use Open Shell to start sb_logger

The most recent events will be displayed and sb_logger will continue to monitor system events as they occur.

Adding Virtual Functions to your SR-IOV Device

To add SR-IOV devices, start at the "Controller" level in the Left-hand menu:

Select the rightmost tab, Controller Actions

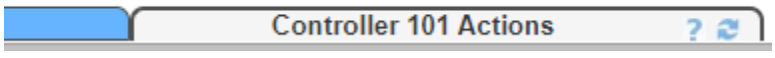


Figure 14: Select Controller 101 then select Controller 101 Actions Tab

And scroll down to NVMe Controller Maintenance

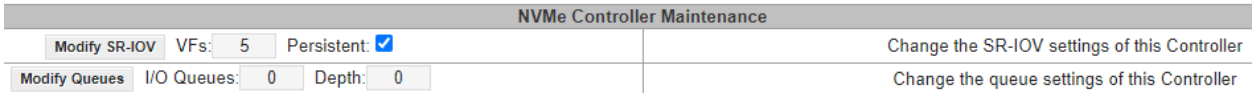


Figure 15: Adding VFs for SR-IOV

Type a number of VFs into the VFs: box, see "5" in the example above, and select the Persistent: box.

Selecting Persistent will cause the Virtual Functions to be available through power cycle or reboot, and therefore discoverable at the BIOS level for the system.

Note: If the Virtual Functions of the SR-IOV device cause issues with the system BIOS or boot path, you will need to physically remove the device from the system before booting if "Persistent" is set.

When you have specified a number of VFs and Persistent (or not), select the "Modify SR-IOV" button.

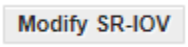
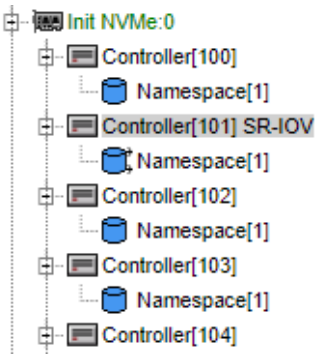


Figure 16: Creating VFs for SR-IOV

After a brief pause, the devices will appear in the left-hand menu:

Before



After

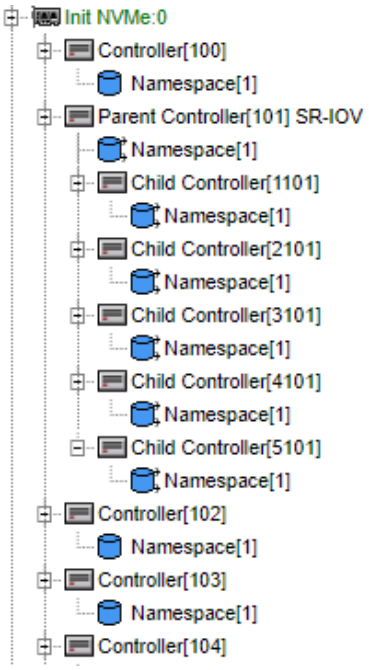


Figure 17: Creation of Child Controllers before and after.

Child Numbering:

Modify the NVMe driver to make target numbers for children (MFND or SR-IOV) be predictable. They will be of the form "child * 1000 + parent", where "child" comes from the "device/function" half of the PCI name (the other half is "bus"). The first child is typically 1, the second child is 2, the third child is 3, etc.

The parent target number is "box * 100 + slot" if the parent is in a SANBlaze enclosure (DT4, RM4, RM5).

In this example the Parent is 101 and we are making 5 child controllers.
child * 1000 + parent 1 * 1000 + 101 =1101 2 * 1000 + 101 =2101 3 * 1000 + 101 =3101
4 * 1000 + 101 =4101 5 * 1000 + 101 =5101

As shown above this is where "Child Controller" 1101 - 5101 are Virtual Functions from the parent device "Parent Controller [101]".

From the CLI enter the following command:

View your Child controllers. Here you can see the 5 Children that were added to the Parent 101. (1101,2101,3101, 4101 and 5101
[root@Qual-DT4-Centos-1-134 ~]# grep TargetIDs= /iport0/port
TargetIDs=100-105,201,1101,2101,3101,4101,5101 (12)
[root@Qual-DT4-Centos-1-134 ~]#
Figure 18: CLI Command to view your 5 Child Controllers

View your controllers nvme list

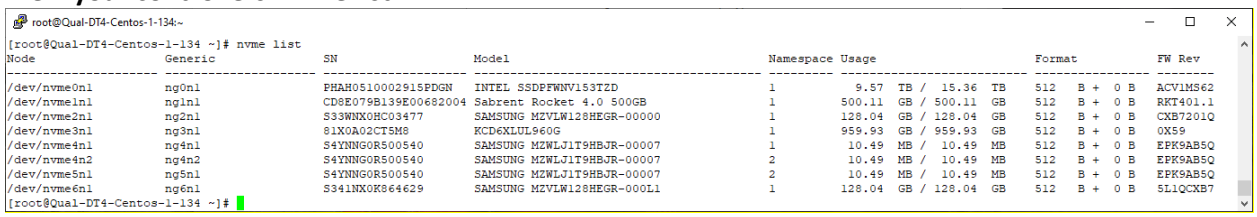


Figure 18: From CLI view of controllers with command nvme list

Creating 5 Child Controllers and make persistent

sbecho NumVFs=5,1 > /iport0/target101

Figure 19: From CLI using sbecho to create 5 Child Controllers and make persistent

From CLI verify 5 Child Namespaces were created using nvme list.

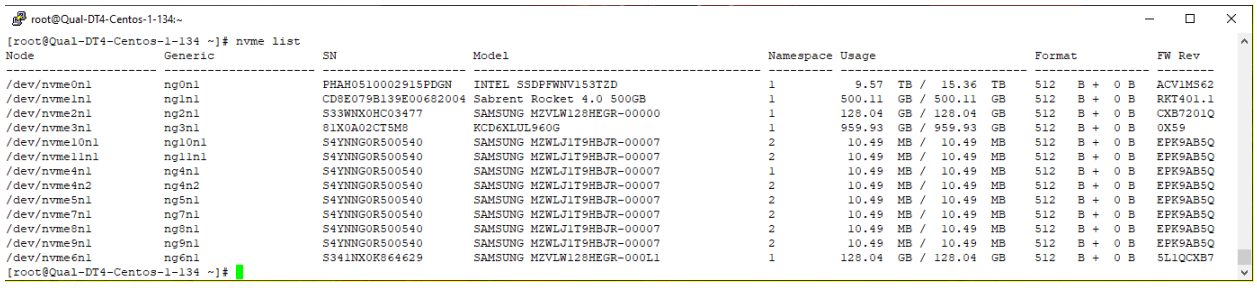


Figure 21: From CLI nvme list shows child controllers were created

If you ever needed to delete the 5 child controllers and their namespaces, you would use this command.

sbecho NumVFs=0,1 > /iport0/target101

Figure 20: From CLI command to delete Child Controllers and their namespaces

Capabilities of Child Controllers

Once the children are created as described above, from a software and testing standpoint they can be treated as if they were physical NVMe devices, with the one important caveat that power and resets of the parent will act on all the children as well.

Note: While SR-IOV children appear as independent controllers, they are children of a single physical device and therefore power and PERST tests will affect the parent and all children.

To see the parent/child relationship and PCIe devices, select a child controller in the left-hand menu:

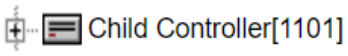


Figure 21: Child Controller and new numbering from Left Hand Menu

And select the Controller Status tab to activate the following display:

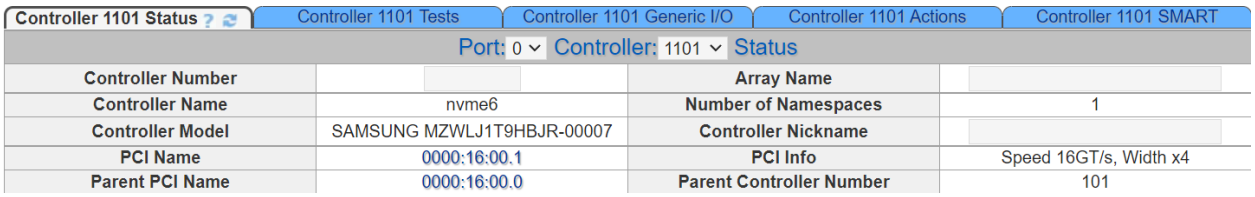


Figure 22: Controller 1101 Status

Note that the selected controller now has a PCI Name at 16:00.1 and a Parent PCI Name at 16:00.0.

View sb_logger as VFs are Added

Returning to the sb_logger window, as the children arrive you will see discovery messages ending as shown below:

```
Jul 27 09:31:55 Qual-DT4-Centos kernel: [ 296.873681] 0000:16:00.1: Controller Probe to Controller Enable took 696594 usec
Jul 27 09:31:55 Qual-DT4-Centos kernel: [ 296.873683] 0000:16:00.1: Controller Enable (CC.EN=1) to Controller Ready (CSTS.RDY=1) took 1296 usec
Jul 27 09:31:55 Qual-DT4-Centos kernel: [  296.874138] nvme_init_identify@3558: 0000:16:00.1: rd 08 00010300
Jul 27 09:31:55 Qual-DT4-Centos kernel: [ 296.874140] nvme_init_identify@3564: 0000:16:00.1: rd 00 001000203c0303ff
Jul 27 09:31:55 Qual-DT4-Centos kernel: [ 296.933000] nvme: enabling affinity for interrupts of device 0000:16:00.1
Jul 27 09:31:55 Qual-DT4-Centos kernel: [ 296.933002] nvme: 0000:16:00.1: setting affinity of IRQ 78 to CPU 3
Jul 27 09:31:55 Qual-DT4-Centos kernel: [ 296.933425] nvme_reset_work@5766: 0000:16:00.1: rd 14 00460001
Jul 27 09:31:55 Qual-DT4-Centos kernel: [ 296.933428] 0000:16:00.1: reset end (succeeded)
Jul 27 09:31:55 Qual-DT4-Centos kernel: [ 296.933463] nvme: adding device 0000:16:00.1
```

Figure 23: Messages from sb_logger when adding children

View of SR-IOV Children from CLI

As the children are discovered by the system, they can be viewed and accessed from the CLI in the same way physical PCIe devices are handled, for example consider the lspci view of the system.

Before creating SR-IOV devices:

lspci |grep -i Non
0a:00.0 **Non**-Volatile memory controller: Intel Corporation NVMe DC SSD [3DNAND, Beta Rock Controller]
0e:00.0 **Non**-Volatile memory controller: Phison Electronics Corporation E16 PCIe4 NVMe Controller (rev 01)
12:00.0 **Non**-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller SM961/PM961
14:00.0 **Non**-Volatile memory controller: KIOXIA Corporation NVMe SSD Controller Cx6 (rev 01)
16:00.0 **Non**-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller PM173X
18:00.0 **Non**-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller SM961/PM961

Figure 24: CLI Command lspci before creating SR-IOV devices

After creating 5 SR-IOV Children

lspci |grep -i Non
0a:00.0 **Non**-Volatile memory controller: Intel Corporation NVMe DC SSD [3DNAND, Beta Rock Controller]
0e:00.0 **Non**-Volatile memory controller: Phison Electronics Corporation E16 PCIe4 NVMe Controller (rev 01)
12:00.0 **Non**-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller SM961/PM961
14:00.0 **Non**-Volatile memory controller: KIOXIA Corporation NVMe SSD Controller Cx6 (rev 01)
16:00.0 **Non**-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller PM173X
16:00.1 **Non**-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller PM173X
16:00.2 **Non**-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller PM173X
16:00.3 **Non**-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller PM173X
16:00.4 **Non**-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller PM173X
16:00.5 **Non**-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller PM173X
18:00.0 **Non**-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller SM961/PM961

Figure 25: CLI Command lspci after creating SR-IOV Children

Note the additional 5 devices at PCIe IDs 16:00.2 - 16:00.6

Viewing and Testing SR-IOV Devices from SBExpress Manager

Select the SBExpress Manager page from the Left-hand menu. If you are already on the page when SR-IOV devices are added, it is necessary to re-select the page after the system configures the VF devices in order to create the Namespace tabs.

Select or re-select from the left-hand menu:

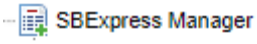


Figure 26: SBExpress Manager from Left Hand Menu

You will notice the Namespace tab for the SR-IOV device has an ellipsis in the tab name, for example below see tab 100.1... The ellipsis indicates that there are child devices on this parent. Select the Namespace tab to show the children.

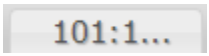


Figure 27: Namespace Tab 101.1

Clicking the 101:1... tab reveals child devices.

Figure 28: Namespace Tab 101.1 shows Children Tabs

The SR-IOV devices for the physical device in slot 1 are:

Controller 101 Namespace 1
Controller 1101 Namespace 1
Controller 2101 Namespace 1
Controller 3101 Namespace 1
Controller 4101 Namespace 1
Controller 5101 Namespace 1

Which correspond to the devices discovered in the left-hand menu:

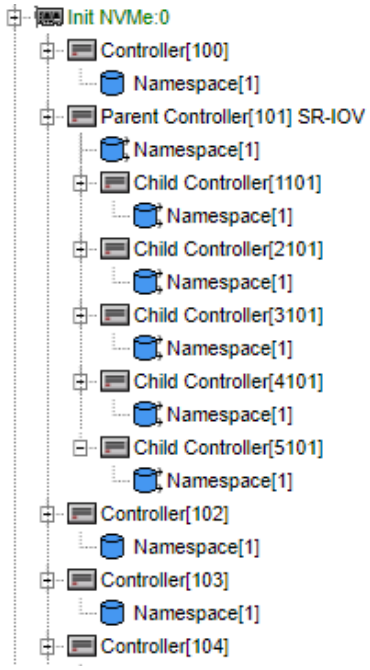


Figure 29: Parent and Child Controllers if Left Hand Menu

Note that if the SR-IOV device is capable of supporting namespaces at the Parent Function (PF), they will be enumerated at the parent, there may be zero or more at the parent level. Similarly, if the device is capable of supporting multiple namespaces on children (VFs) they will be enumerated at the child controller level.

Each namespace will receive a tab on the SBExpress page, which can be used to access the namespace to assign tests, as shown in the following section.

Note: When changing the SR-IOV configuration by adding or removing virtual functions, it is necessary to refresh the "SBExpress Manager" once to pick up the change in device configuration. After the page correctly reflects the new configuration, it should not be necessary to refresh the page manually. Click here to refresh the page:

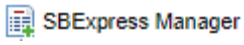


Figure 30: Refresh SBExpress in Left Hand Menu

Assigning Certified by SANBlaze Tests to SR-IOV Devices

Once your system has discovered and configured the SR-IOV Physical and Virtual functions as described above, you may now test the virtual functions in the same way as physical functions as described below:

Starting on the SBExpress Manager page with tab 101:1 selected (PF), assign tests to namespace 1 on the parent:

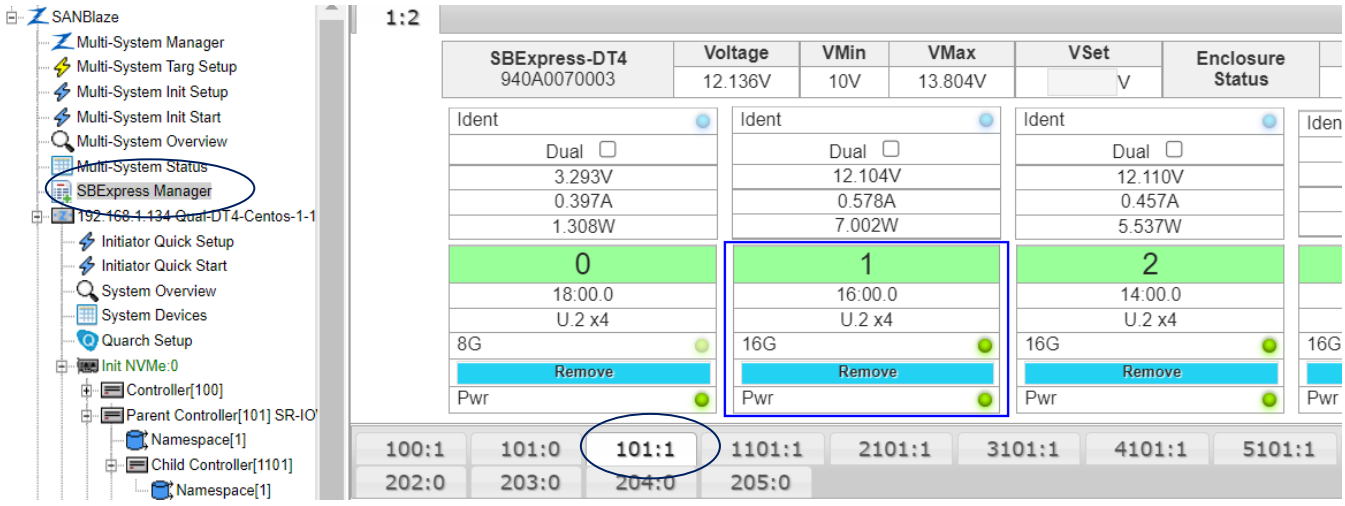


Figure 31: Selecting a Namespace from SBExpress Manager

Assigning Certified by SANBlaze Tests

Scroll to the bottom of the page, locate the "Certified by SANBlaze" test section, expand the sections containing the tests you wish to assign:

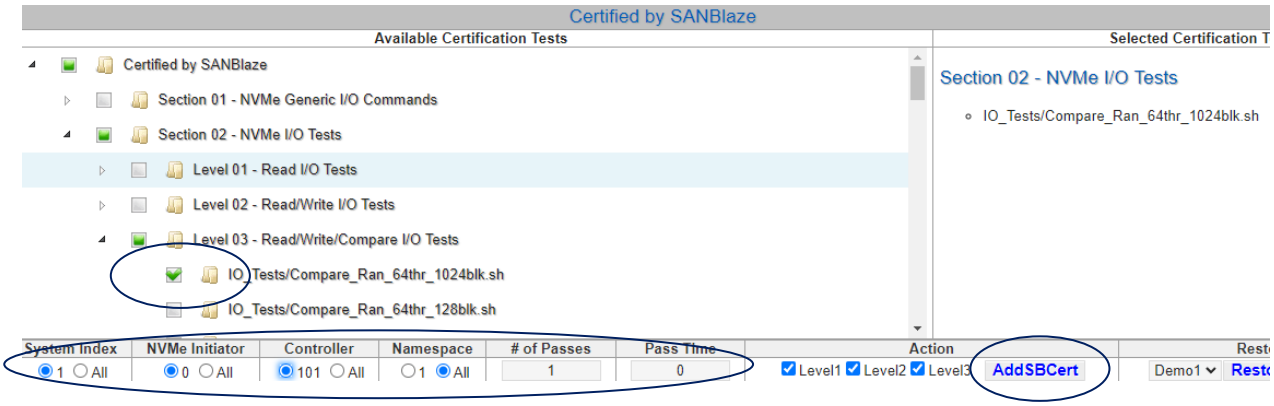


Figure 32: Assigning SBCert Tests to a Selected Namespace

For the purpose of example I have selected one test "IO_Tests/Compare_Ran_64thr_1024blk.sh" and I have selected the scope to assign the test to as "System 1, NVMe 0, Controller 101, Namespace All, Passes 1 and Pass Time 0 (Pass time 0 will allow the system to assign the test time).

Note also that I have selected Level 1, Level 2 and Level 3 tests. This will make the entire list of tests available.

Once you have selected the tests, scope, level and duration of testing as above, select the AddSBCert button to assign the tests to the device.

The page will update after assigning the tests, you can start the tests at this point using the "Start" button as shown below:

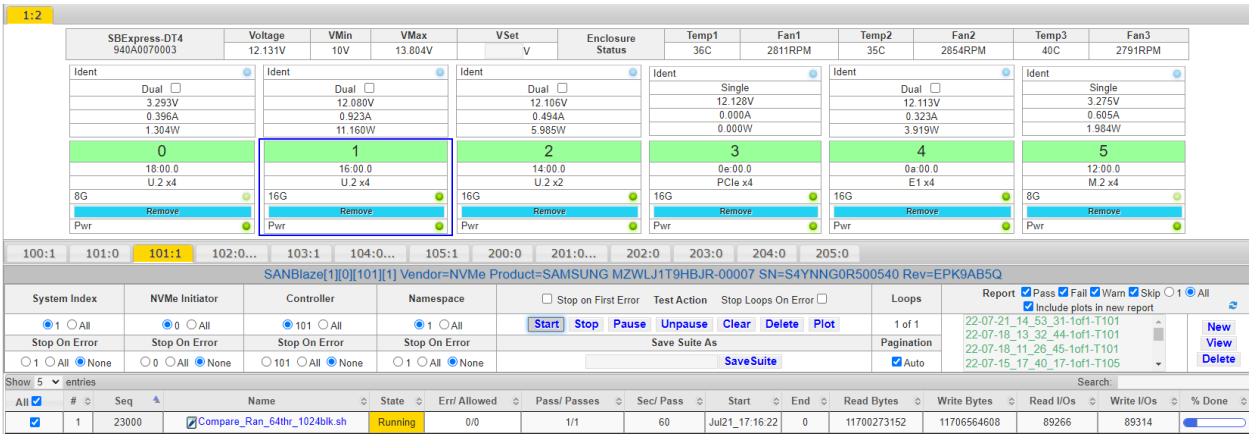


Figure 33: Running SBCert Tests on a Selected Physical or Virtual Function

Building a Test Suite for the Virtual Functions

Using the method described above, you can add one or more tests to a range of Namespaces. To simplify this process, SBExpress allows building a group of tests into a test "Suite" and subsequently assigning the suite to a namespace or range of namespaces as shown in the example below.

Starting with the configuration above with one test assigned to the Parent only, we'll take the following steps:

- Clear the status of the existing test (optional)
- Add 3 more tests to our test suite and name it SR-IOV_Children
- Assign the suite to each of the children
- Start testing on all devices at the same time

Clear the status of the existing tests

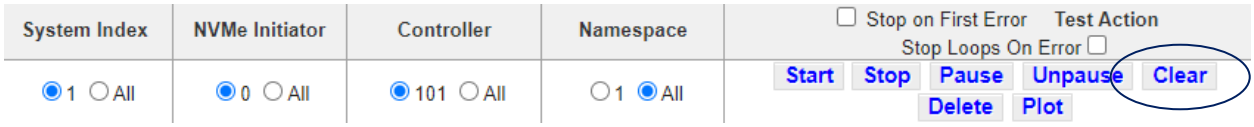


Figure 34: Clearing status of existing tests

Add Additional Tests to our Suite

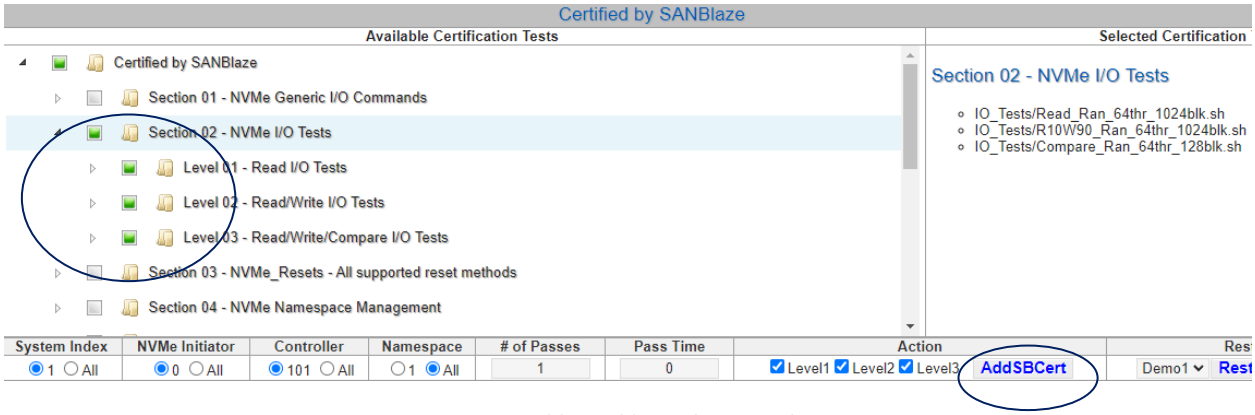


Figure 35: Adding additional tests to the suite

Build a Named Suite from Selected Tests

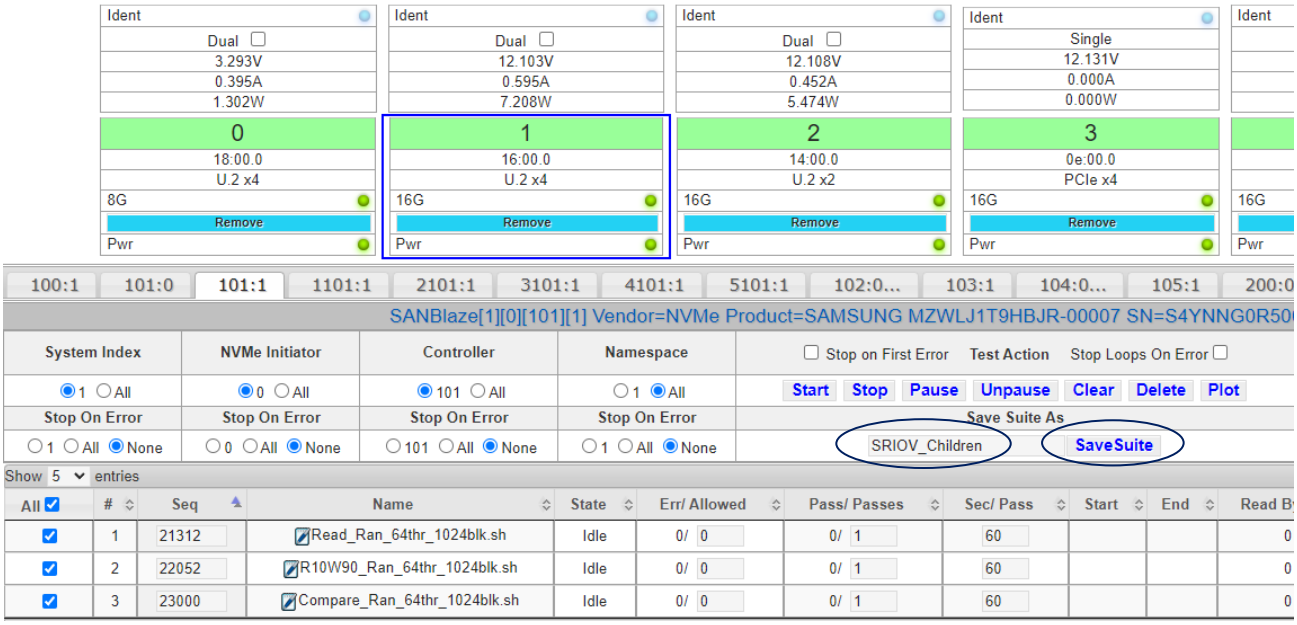


Figure 36: Saving a Suite named SRIOV_Children

Selecting the "SaveSuite" button will create a named suite of tests that can then be assigned to the children.

Assigning the Saved Suite to the Child Devices

Once the suite has been created as in the example above, it can be assigned to one or more children using the procedure below.

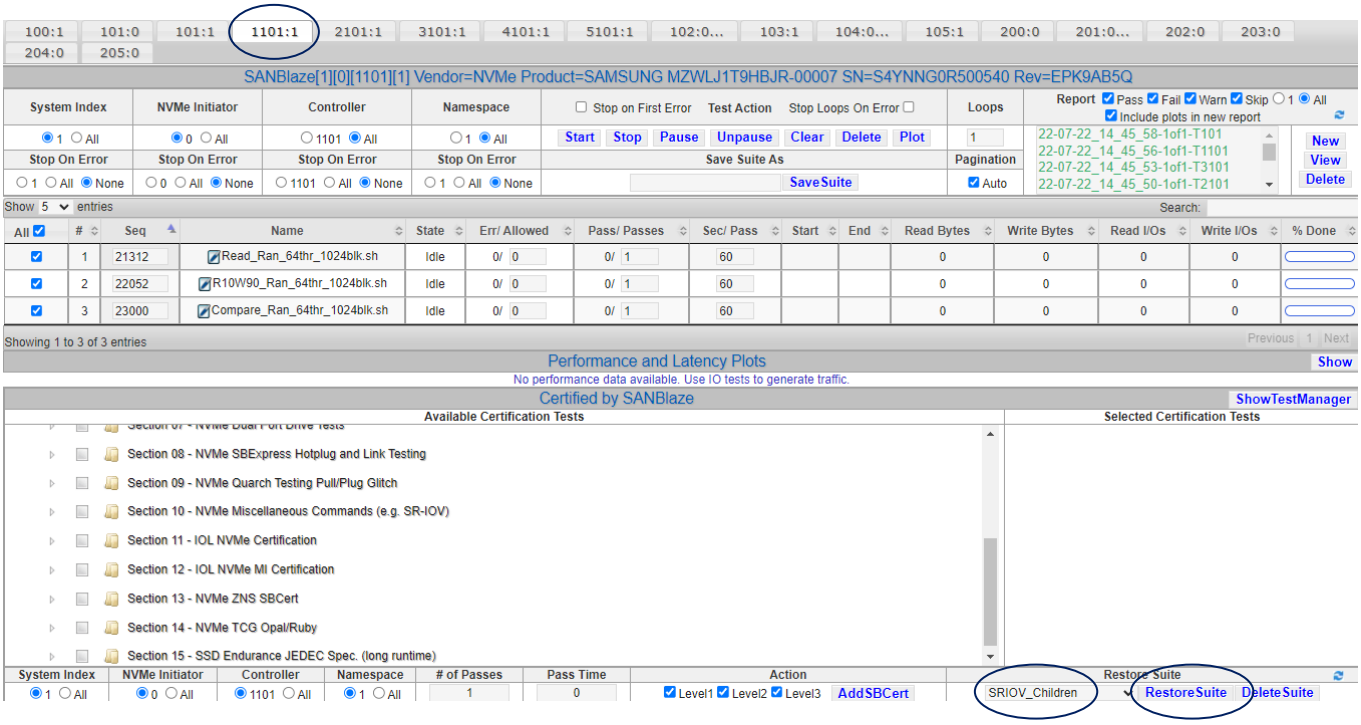


Figure 37: Restoring a Suite named SRIOV_Children

Target 1101: Namespace 1 tab selected, Select the SR-IOV_Children suite and select the "RestoreSuite" button. As seen in the example above, the tests are assigned to the child.

Note: As of the date of this writing, it is not possible to assign tests to all children of a given parent automatically. This feature is planned for a future release. For now, the tests or suite must be assigned using the method above to each child.

Running Tests on All Children

Once tests have been staged to the parent and all children, they can be started and monitored together.

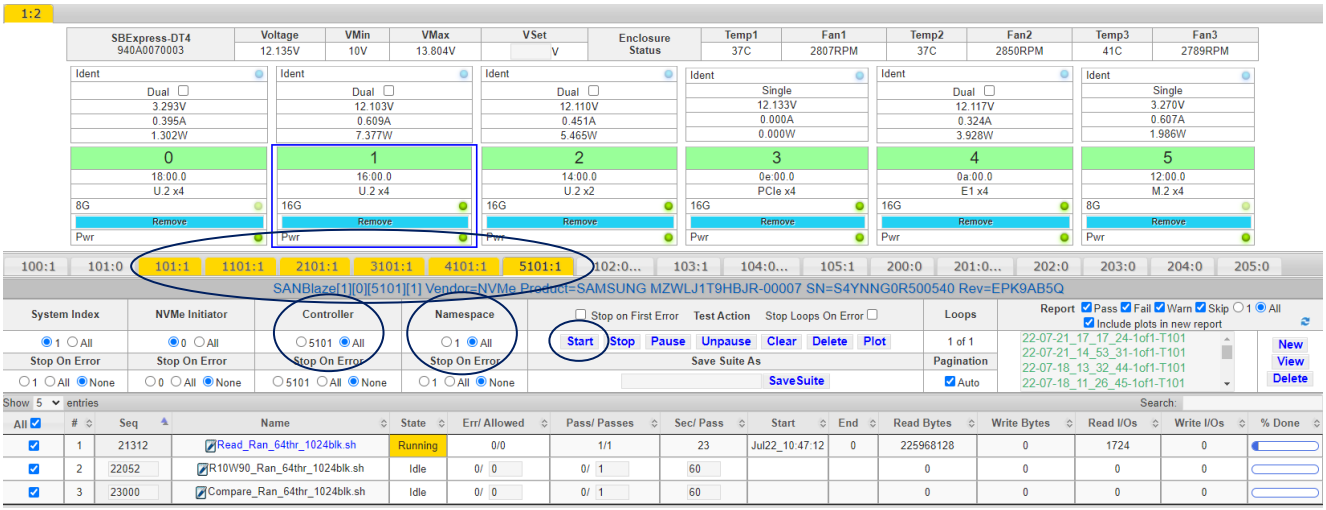


Figure 38: Starting a Test on Parent and all Child Controllers

Select all Controllers and all Namespaces, then Start. Tests will begin on all controllers within the selected scope of devices. The status of tests is reflected by the color of the namespace tabs. If any test on a namespace fails, the tab will change color to reflect the status.

Running Test Reports

Once you have assigned and run one or more tests, reports are automatically generated and can be viewed, saved and exported using the "View" button as in the example below.

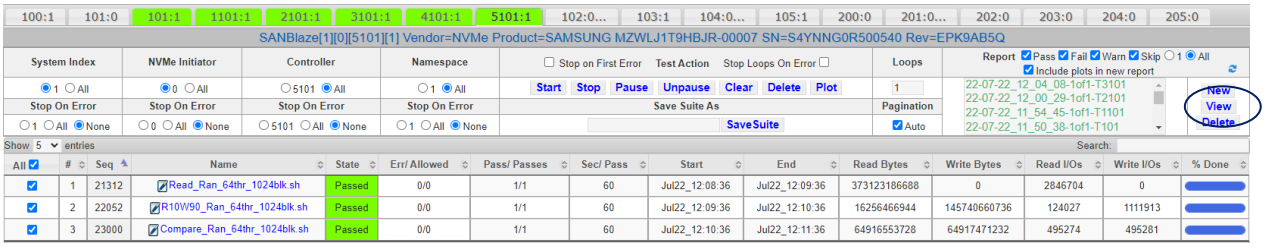


Figure 39: Viewing the Test Reports

Viewing the Test Report

The test report view will show an overview of the tests recently run and will allow for accessing full details on one or more tests, save a self-contained interactive version of the test report and also generate a printable version of the report.

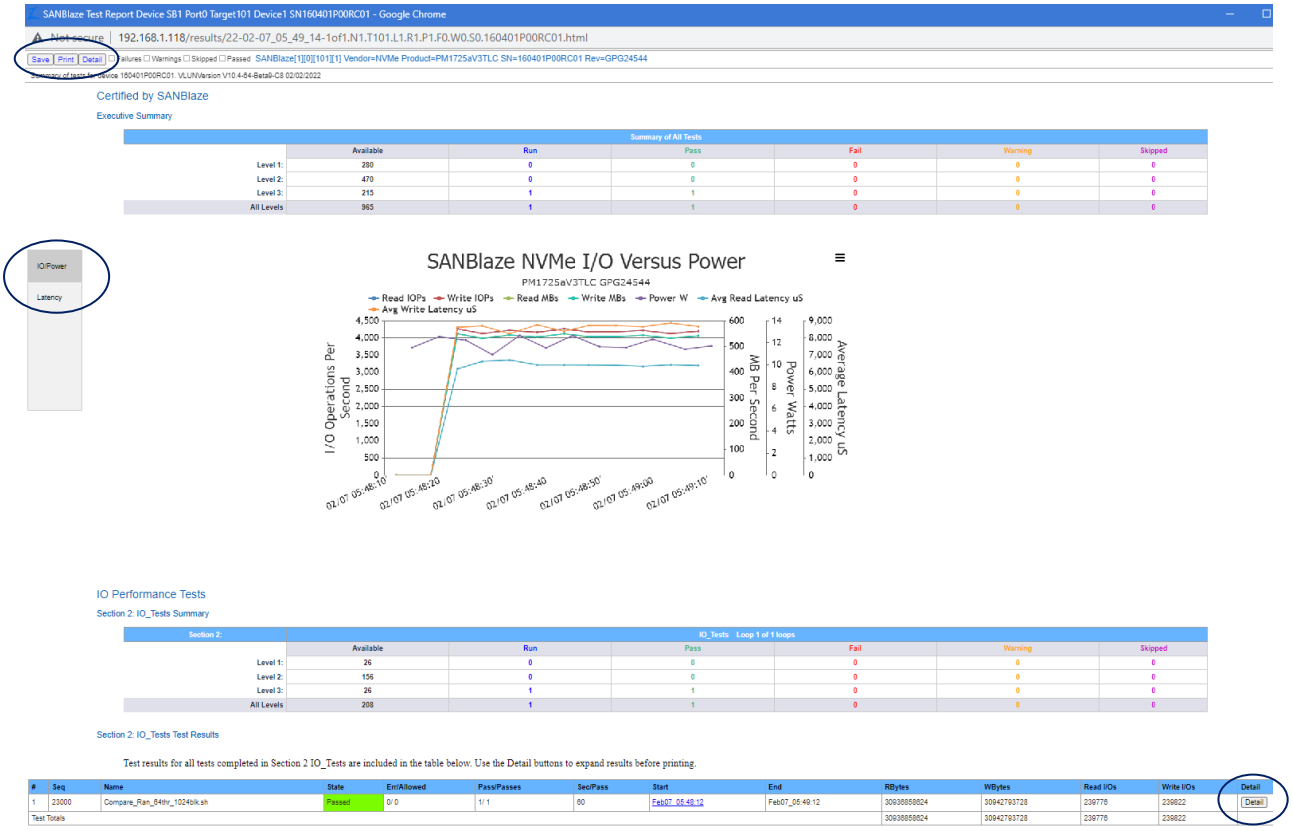


Figure 40: Printable Version of test report

Note that performance plotting has been added to reports. The graph show above is interactive, allowing zooming of any given test time. Latency plots are also included in the report.

- Print - Pretty print the report
- Save - Save a complete self-contained, interactive copy of the report (html/JavaScript)
- Detail - At the top of the page will expand test detail for selected status (skipped, failures, warnings).
- IO/Power or Latency - Select which type of plot to include in report
- Detail - On each test line will expand details on just one test

Note: Test reports are generated automatically as each test suite completes on a per-namespace basis. To archive all tests for an SR-IOV and all children it is necessary to save reports for all children of a given controller.

SR-IOV Testing for Dual Path Devices

The SANBlaze SBExpress Gen5 and Gen4 systems dynamically support testing of single and dual port devices on both U.2 and EDSFF device types. If the SR-IOV device supports Dual Path, you may change back and forth from single port to dual port operation as shown below.

Select the SBExpress page, and locate the "Dual" checkbox for the given device, in the example below slot 1, parent target 101.

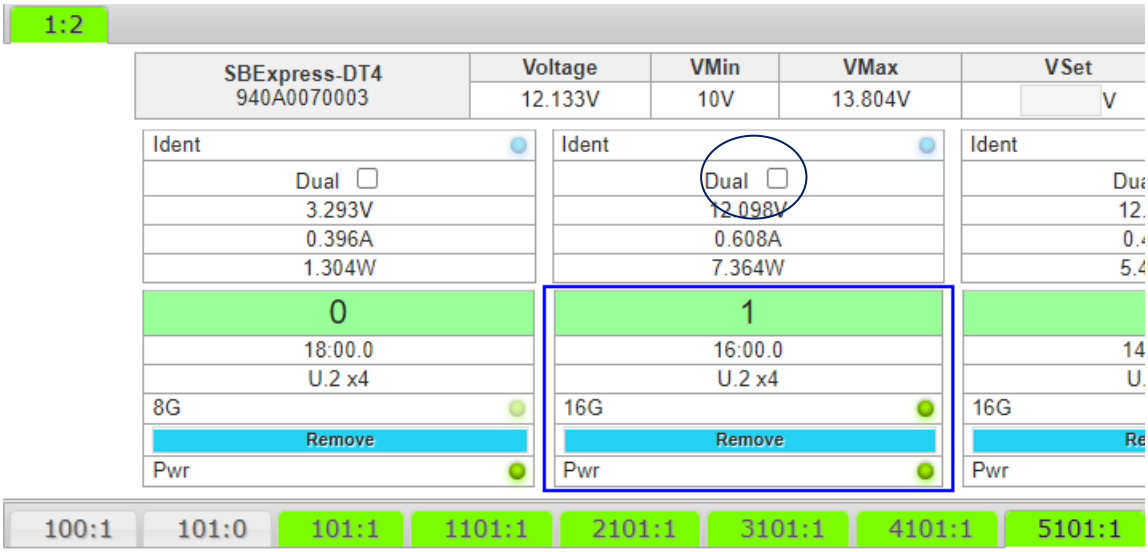


Figure 41: Check off Dual

To enable Dual Path testing, select the "Dual" checkbox shown above. You will see the following:

- The slot will switch from single port U.2 1x4 to U.2 2x2 PCIe connections
- Additional target numbers will be added, in our case Target 201 to point to the "other" port
- The Left-hand menu will update to reflect the change
- Additional Namespace tabs will be created to allow testing of the second port devices

Note*:* You must reload the SBExpress page to build the additional tabs by clicking here:

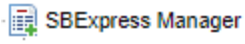


Figure 42: Select SBExpress Manger to reload page

When the changes are complete, you will see these changes reflected on the SBExpress Manager page.

Left-hand menu Displaying Dual Path SR-IOV Parent and Children24

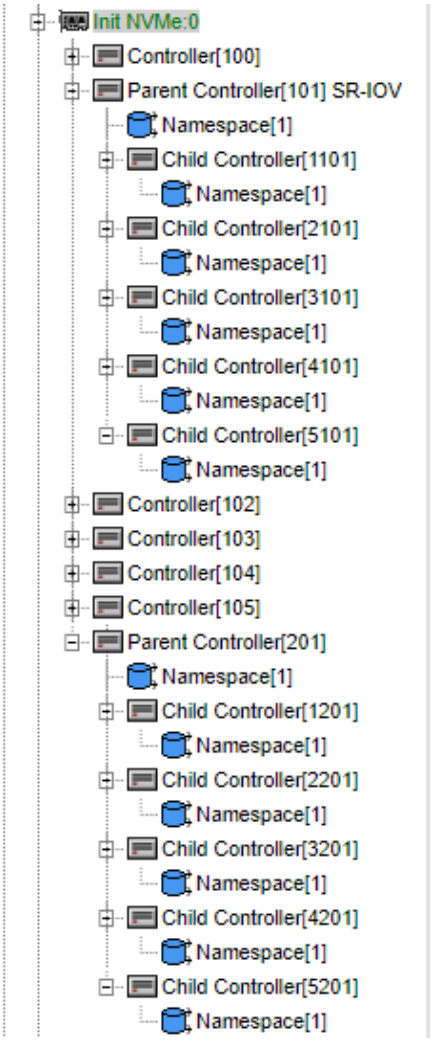


Figure 43: Shows Dual Drive Controller results after selecting Dual

Note the graphical representation of a multi-path function has two arrows, indicating there is more than one path into the device.

Selecting any namespace and the status tab:

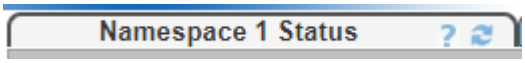


Figure 44: Namespace 1 Status Tab

Will provide a view of the available paths to the device:

Figure 45: Results of Namespace 1 Status Tab

SR-IOV Specific Testing Considerations

Once your SR-IOV targets are created and exposed to SBExpress Manager, you can assign tests and run reports in exactly the same way as if they were physical devices but with some very important considerations.

- PCIe Resets (PERST) will reset the parent and ALL children (so tests on children may fail if not expecting a reset) on a given PCIe path
- Power for the slot affects both paths and all children
- Any test that creates and destroys children must take into account testing running on child devices

Special Namespaces and the Zero Namespace

Namespaces in NVMe index from 1, therefore the lowest real namespace on an NVMe system is by specification Namespace1. SBExpress will create a "placeholder" namespace at Namespace0 in the case where there are no namespaces at all on an NVMe device.

The purpose of this special Namespace0 is to stage tests which will test functions of a controller not requiring a Namespace to execute I/O to. Any SANBlaze test that touches controller functions only but NOT namespaces can be staged to run on Namespace0. You can think of Namespace0 as "Controller Namespace".

A specific use case for Namespace0 would be a test that creates namespaces on a device that has none, or creates or destroys children on an SR-IOV capable device.

If there is no Namespace0 on the SBExpress page for a given controller, you can stage controller-based tests to Namespace1 or any other Namespace.

- Namespace0 - Can only run Controller tests
- NamespaceN - Can run Namespace tests and Controller tests

Appendix A: FAQs

Below are a few questions and answers that may be helpful.

Q: I created a lot of Virtual Functions under a controller. I would like to stage and execute tests on the parent and all children on the one physical device. How is this done?

A: In release 10.5, tests must be added to each child. There is no "All Children of this Parent" scope at this time.

Q: Given the fact that there is no "All Children of this Parent" scope, do I have to add each test to each child individually?

A: No. You can use the "Suite" functionality to build a group of tests into a suite, then apply the suite to each child. Example is given in the document.

Q: I don't see the "Modify SR-IOVs" functionality on my Controller Actions Page at all. What am I doing wrong?

A: Nothing, but if the "Modify SR-IOVs" dialog does not appear on page, the selected target does not support SR-IOV.

Q: I selected the "Remove" button on the SBExpress Manager page and lost the parent and all children. Why?

A: The "Remove" button removes the NVMe device at the parent level, and therefore all children are also removed, yes.

Q: Why are you showing a Namespace0 when NVMe Namespaces start at 1

A: The Zero Namespace is a staging area for devices with no namespaces, which can be used to run controller only tests, specifically for tests that create and destroy namespaces or children.