INFINIDAT

TECHNICAL REFERENCE

InfiniSafe® for InfiniBox® Reference Architecture Guide

Delivering Cyber Resilience

Version 1.1

Table of Contents

Introduction

Infinidat believes having a multi-tiered strategy for cyber resilience is very important to a business and how they further extend their data protection capabilities should you encounter a cyberattack event. In February of 2022, we introduced InfiniSafe for our InfiniGuard® purpose-built backup appliance. The InfiniSafe Reference Architecture extends our cyber resilience capabilities to our InfiniBox family of products. Today, protecting the business from the likelihood of experiencing impact from a cyberattack or ransomware event is a top-of-mind-concern. In the Fortune 500 2021 survey of CEOs, the #1 threat to their businesses cited was cyber security, and in a 2022 IDC survey of Board of Director members the #1 concern was also cyber security. In fact, 86% named it the #1 concern for their companies. With this, Infinidat has defined our InfiniSafe frameworks around our ability to help companies create a multi-tiered cyber resilience strategy across our family of products.

InfiniSafe Architecture Overview:

The Infinidat InfiniSafe Reference Architecture for InfiniBox responds to four critical areas of data security that are important to developing a cyber resilience solution. Each of these elements are independent of each other and can be implemented individually or collectively to create the desired level of security that aligns with your organization's business protection goals. This guide will outline and define each of these elements, and provide both a preparing aspect to a cyber Incident as well as recovery from a cyber incident.

1. **Immutable Snapshots** - *Required foundational cyber security feature*
2. **Near-instantaneous Recovery** - *Required foundational cyber security feature*
3. **Fenced Network Forensic Environment** - *Suggested cyber security feature*
4. **Remote Logical Air Gap** - *Optional cyber security feature*

Immutable Snapshots *(Required)*

This ability applies directly to the creation of numerous point in time copies of data in the most efficient manner possible. Snapshots are easy and efficient. InfiniBox easily creates immutable snapshots that provide absolute security of the data. There is ZERO ability for that to be changed or altered and snapshots can only expire based on the retention policy defined in order to be removed.

Immutable snapshot frequency should be automated to occur at least daily for all primary storage volumes. Higher business application volumes such as databases and email servers are recommended to occur hourly, if not every quarter of an hour. The frequency of InfiniBox immutable snapshots has no negative impact on the production application performance with capacity impacts restricted to modified blocks only. Utilizing more frequent immutable snapshots will allow for more granular point-in-time database images, less data loss, and shorter archive log roll-forward time.

InfiniBox immutable snapshot retention suggests being retained for 7 - 14 days, however Infinidat encourages your organization to assess if a larger number of snapshots will reduce your business application risk. This is a practical range of cyber data protection providing multiple points in time of restoration, balancing speed of recovery and capacity required of snapshot change rate. Infinidat provides a simple method of immutable snapshot scheduling and retention management via a feature called SnapRotator. Infinidat SnapRotator resides external to the InfiniBox array running on a hardened Linux OS at your organization's security standards. Businesses should also implement immutable snapshots within their secondary storage used for backup application repositories. Infinidat's cyber protection solution for secondary storage is InfiniGuard® which leverages InfiniSafe technology as well. The full SnapRotator User Guide can be accessed using the following URL.

https://support.infinidat.com/hc/en-us/articles/360003909237-InfiniBox-SnapRotator-User-Guide

Near-instantaneous Recovery *(Required)*

InfiniBox data restoration uses its InfiniSnap® technology. InfiniSnap immutable snapshots utilize a redirect on write methodology, as opposed to a copy on write methodology, for non-impacting instantaneous access to the locked data within the immutable snapshots. Returning to normal business operations is the most critical aspect in a cyber incident. Paramount to achieving this is recovering your production data back to the closest point in time just prior to the malicious code being executed. Secondarily to a rapid business recovery is validating your immutable snapshot data and knowing what is deemed to be "clean."

Fenced Network Forensic Environment *(Suggested)*

Creating an InfiniBox fenced environment consists of isolation of needed secure server resources that would then connect to a private network via FC or Ethernet within InfiniBox. Access to this private network can be dynamically created via API automation or manually created using InfiniBox GUI by an authorized storage administrator. These fenced secure server

resources can be granted access to the immutable snapshot dataset copies either on the local InfiniBox or on a remote air gap InfiniBox immutable dataset copy. Practical use cases for leveraging the fenced network are twofold; attaching a secure server to the immutable copy datasets for forensic scanning by an organization's security applications is the first function. Rehearsing a cyber recovery event, such as a complete Oracle database server recovery, not unlike traditional DR tests, is the next functional use of the InfiniSafe Fenced Network.

Whether using forensic scanning tools or cyber recovery rehearsals inside the Fenced Network, a company's organization can know exactly which points in time are valid immutable snapshots and available to recover from if needed. In the industry this clean point in time copy of datasets is called a "known good copy" of the data. Assessing and identifying the known good copy of your immutable snapshots requires a defined and routine process. At a minimum, the critical business applications should have forensic scanning of their immutable snapshots on a bi-weekly basis. Once these immutable snapshots are in a routine schedule of locking and expiration based on your organization's desired security policy, augmented with scheduled forensic scanning, the ability to confidently recover from a cyber event would be certain and pass many of the upcoming compliance audits enacted by most organizations' risk departments.

Remote Logical Air Gap *(Optional)*

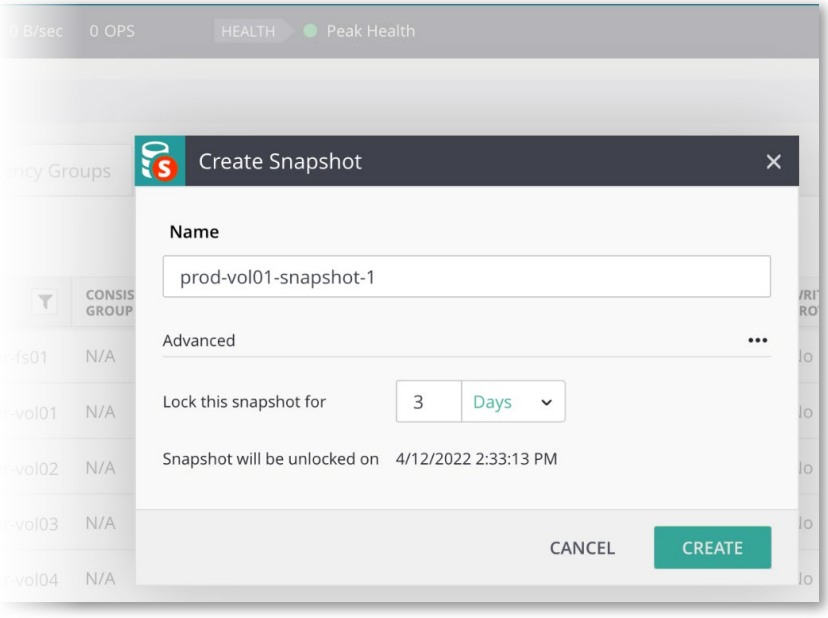
InfiniBox air gap provides organizations with the ability to identify datasets within their InfiniBox that they deem critical and need to create a cyber resilient external copy strategy for it. Infinidat's InfiniSafe technologies and associated reference architecture provide our customers with a solution that enables them to remotely replicate those datasets to another InfiniBox located within another environment and create immutable copies of the replicated datasets on the target InfiniBox, thereby creating an air gap protected copy.

Begin InfiniSafe For InfiniBox Preparation Stage:

As stated within the overview section of this guide, InfiniSafe elements of InfiniBox can be executed manually via the InfiniBox GUI or via API automation such as Ansible. The next portion of this guide will present the suggested chronological steps to enacting the four key elements of the InfiniBox InfiniSafe Architecture along with illustrations from the InfiniBox GUI. As of the time of this document publication, there are Ansible modules that align with each phase of the reference architecture for automation purposes. There are additional methods and tools such as InfiniShell or InfiniSDK which can be instrumented to automate all of the features defined within this reference architecture. These automation frameworks and the associated documentation on leveraging the preformed modules can be found on the downloads section of support.infinidat.com for authorized Infinidat customers.

PHASE 1: Creating Immutable Snapshots

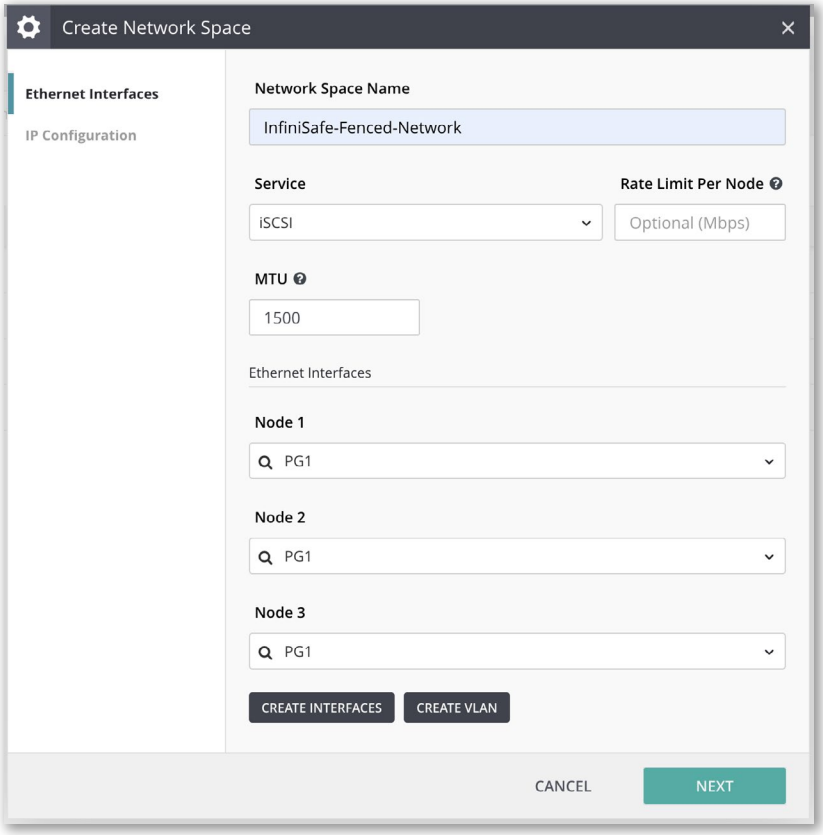
Snapshots on InfiniBox can be created on a volume, consistency group (a set of selected volumes) or a filesystem. The illustration at right provides a visual example of the core snapshot creation action via InfiniBox GUI. In this example we are creating an immutable snapshot for 3 days on a single volume. The snapshot name is prod-vol01-snapshot-1 and at the time of creation the InfiniBox will advertise the date and time the snapshot will be unlocked as referenced prior to creation.



Appendix A of this guide will provide a step by step procedure using the InfiniBox GUI, InfiniShell commands for scripting, and Ansible API modules for your reference.

PHASE 2: Creating a Fenced Network Forensic Environment

InfiniBox is a unified storage platform supporting Fibre Channel, iSCSI, NFS and SMB. The network access to all ethernet protocols is based on a virtual abstraction of the network transport protocols for performance, resiliency and consistency. Creating a fenced network with the ethernet protocols will occur by temporarily creating an InfiniBox "Network Space" to present one or a series of immutable snapshots created earlier in this guide. The image below will illustrate how a storage administrator could manually create an iSCSI network space. This iSCSI network space would be an isolated transport for any immutable snapshots created on the InfiniBox.



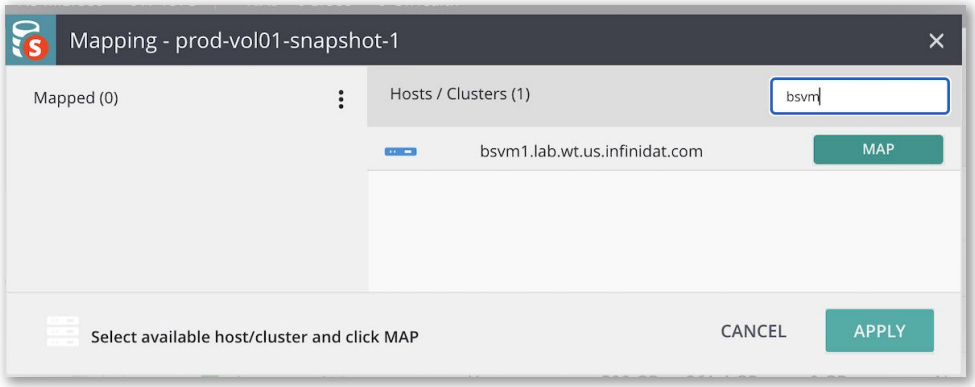
PHASE 3: Map Immutable Snapshot to iSCSI Fenced Host

Step 1:

Mapping the immutable snapshot to a host connected to the recently created iSCSI Fenced Network would be the first step to identifying the known good copy. Mapping the immutable snapshot directly would only permit read only actions such as forensic scanning by 3rd party tools such as Malwarebytes, Crowdstrike, SentinelOne and numerous other OEM providers. The image below illustrates the immutable snapshot "prod-vol01-snapshot-1" to host bsvm1.lab.us.infinidat.com which is a Linux VM Guest that has an iSCSI connection to the InfiniBox Fenced iSCSI network.

Step 2:

After the forensic scan has been completed confirming the snapshot is clean, your cyber security scanning process should include amending the snapshot name to include a notation validated "clean."



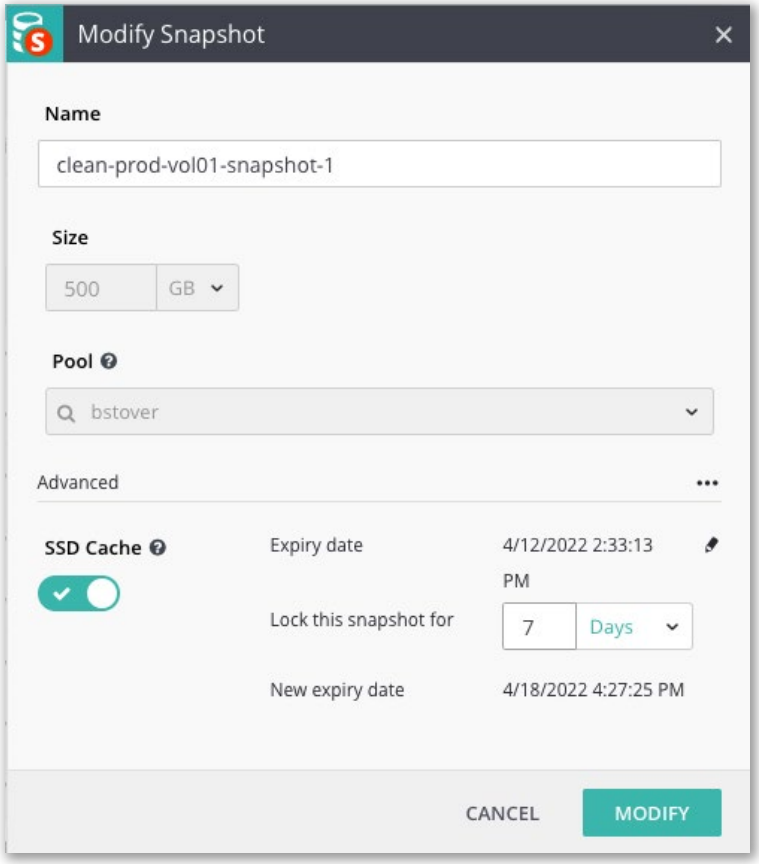
The next logical step would be to rehearse your application recovery. The InfiniBox storage admin would create an additional snapshot of the immutable snapshot deemed as clean. This would be a snapshot of a snapshot which has a hierarchical association to the primary volume but without any performance impact. Using the 2nd non-locked snapshot just created from the immutable snapshot of "prod-vol01-snapshot-1" the storage admin should change the non-locked snapshot from a read only attribute typically assigned to InfiniSnaps to a read/write attribute. Then map the non-locked snapshot to the rehearsal host. This would be the same process as mapping in step 1 except the snapshot is able to be accessed in a writable fashion to rehearse an application recovery such as an Oracle database instance restart with subsequent transaction log replay to bring the database to current state just prior to the fictitious Cyber incident.

End InfiniSafe For InfiniBox Preparation Stage

Begin InfiniSafe For InfiniBox Cyber Incident Recovery Stage:

PHASE 1: Extend the expiration date of impacted immutable snapshots

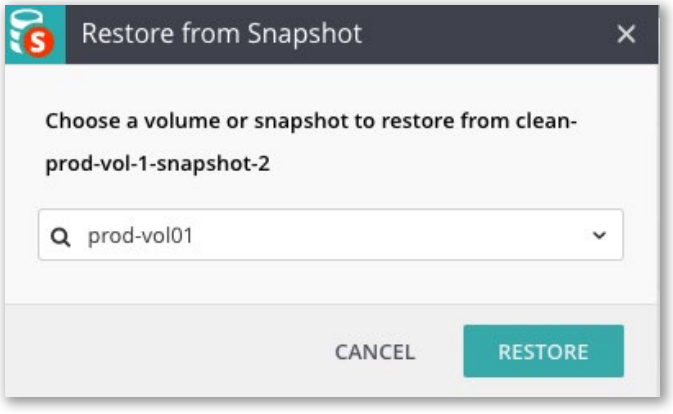
When a Cyber incident is discovered, the storage admin should extend the lock on the last "known good copy snapshot" plus all subsequent immutable snapshots until business operations have been restored. The Ponemon Institute in 2020 found the mean time to containment of a malware event for the average organization is 73 days. Therefore if a cyber event such as malware were to infect an application or portion of your company's infrastructure, extending the locked duration on a "known good copy" of the impacted datasets is a key first step. Your organization will not likely take 73 days to remedy the situation, but your organization does need to be certain the validated copy of secure data remains unalterable throughout the recovery and validation phase. Extending an immutable snapshot life is a simple modification of the snapshot metadata. In the illustration below the storage admin has located a previously labeled clean immutable snapshot which has been scanned and marked as a viable candidate for a restore, the snapshot name is "clean-prod-vol01-snapshot-1". The original expiration date is " 4/12/2022 2:33:13 PM," however the storage admin is about to configure the expiration date to extend for an additional 7 days. This immutable snapshot will be locked until 4/18/2022 4:27:25 PM."



PHASE 2: Recover production dataset from immutable snapshot

The InfiniBox immutable snapshot can be used as a restore object that contains the protected datasets for the entire application regardless of the snapshot position in the protected copy chain. Prior to issuing the immutable snapshot restore to the production volume, take a manual snapshot of the production volume. This manual snapshot may be used for further forensic analysis or as a backout step from the pending immutable snapshot restoration process below.

In this example we will use "clean-prod-vol01-snapshot-2" which is the second snapshot in the protected datasets of "prod-vol01". While this example is going to restore the production volume "prod-vo01" we could also roll back an unlocked snapshot in the snapshot recovery chain if that behavior was desired.



It should be noted that different operating systems may require the primary volume to be unmounted or even unmapped from the host prior to the storage admin prod-vol01 restore operation. Please consult your operating system best practices.

End InfiniSafe For InfiniBox Cyber Incident Recovery Stage

Summary of InfiniSafe Reference Architecture for InfiniBox

Infinidat's response to cyber security challenges is a simple, three phase approach. Protecting datasets via point in time locked copies of your critically defined elements. Regardless of local copies or remote air gap copies, the ability to protect using as many copies as possible with unique retention durations is made available. InfiniBox enables your business to rehearse and validate that your protected copies of data are indeed spyware or virus-free utilizing the InfiniBox immutable snapshots inside an isolated fenced network environment. The InfiniSafe features are consistent on both InfiniBox or InfiniBox SSA thereby enabling cost efficiencies for leveraging production copies on the most performant all flash platform InfiniBox SSA, while driving down costs on the remote protected copies using an InfiniBox hybrid array without a technology change or process change which would increase operational risk.

Appendix A

The following outlines the step by step procedures in logical order of operations for the InfiniSafe security elements defined in the InfiniSafe Reference Architecture for InfiniBox using InfiniBox GUI or InfiniShell CLI. Consult your Infinidat Technical Sales Engineer if you encounter a step needing assistance.

PHASE 1: Protection - Immutable Snapshots
Method A - InfiniBox GUI
Method B - InfiniShell with SnapRotator

PHASE 2: Validation - Fenced Forensic iSCSI Network
Method A - InfiniBox GUI
Method B - InfiniShell

PHASE 3: Replicas - Remote Logical Air Gap
Method A - InfiniBox GUI
Method B - InfiniShell

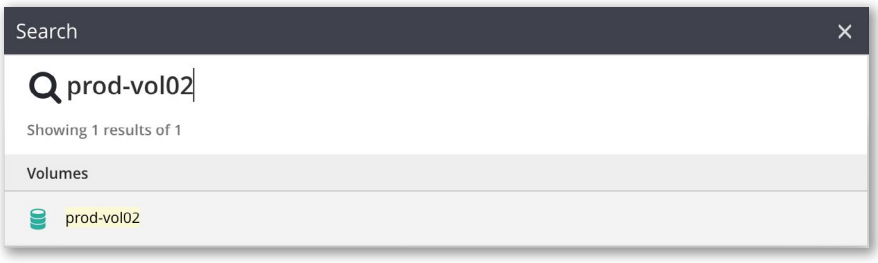
PHASE 4: Recovery - Immutable Snapshot Restoration
Method A - InfiniBox GUI
Method B - InfiniShell

PHASE 1: Protection - Immutable Snapshots

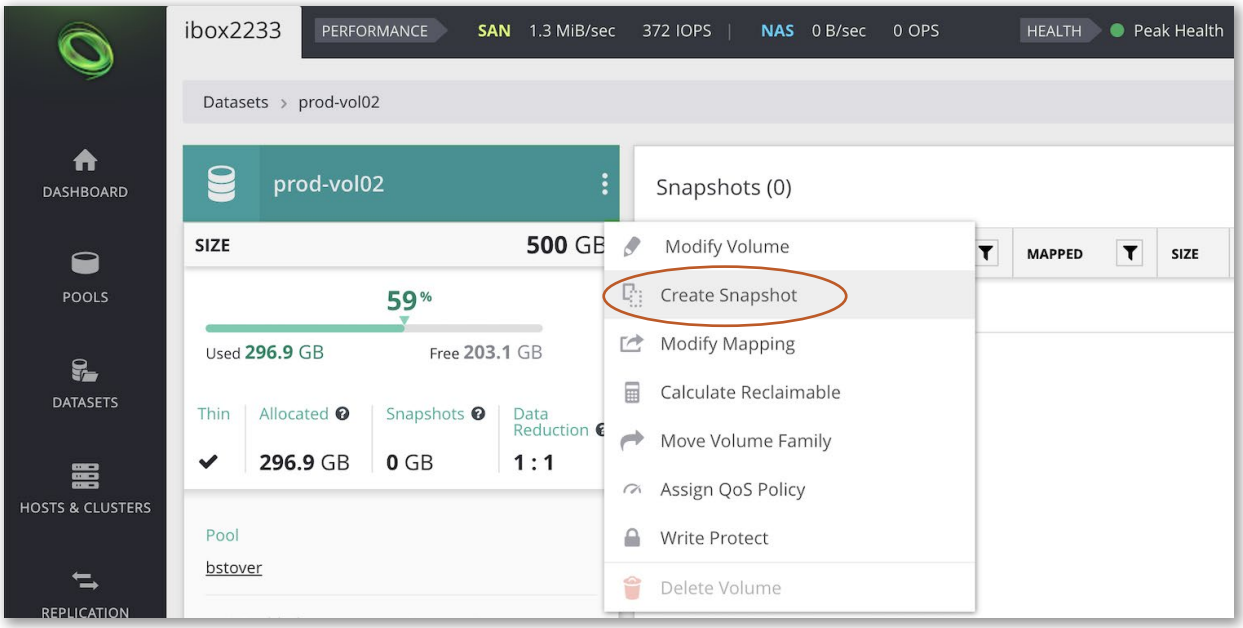
METHOD A: InfiniBox GUI

The next few steps will outline step by step how an InfiniBox storage admin would manually create an immutable snapshot on a primary volume utilizing the InfiniBox GUI.

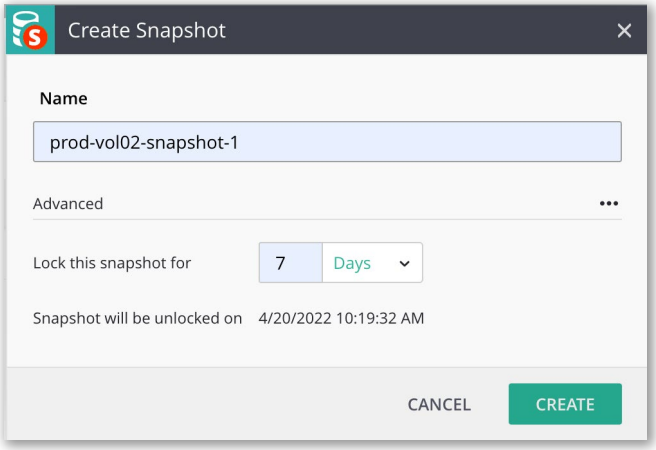
Step 1:
Locate the desired volume, this can be accomplished by using the general search window in the upper right hand side of the main InfiniBox dashboard. In this example we will search and select "prod-vol02"



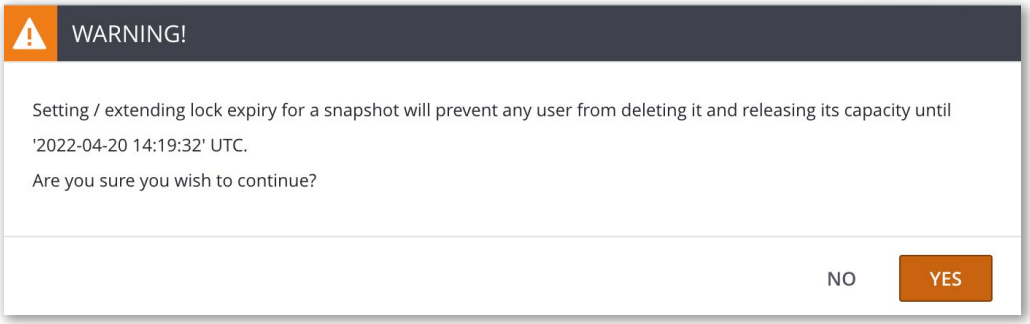
Step 2:
Right click on the prod-vol02 object using the 3 vertical dots to select Create Snapshot



Step 3:
Create snapshot with 7 day lock to create immutable snapshot, select the 3 horizontal dots to access the Advanced feature of the Create Snapshot.



Step 4:
Acknowledge the warning about when the lock will expire before the Create Snapshot action completes.



Step 5:
Validate the expiry date is 7 days out by reviewing the snapshot creation column vs snapshot expiry date column.

NAME	CONSISTENCY GROUP	MAPPED	SIZE	ALLOCATED	SNAPSHOTS	LOCK STATE	LOCK EXPIRY DATE	CREATION DATE
prod-vol02-snapshot-1	N/A	No	500 GB	0 GB	0 GB	LOCKED	4/20/2022 10:19:32 AM	4/13/2022 10:23:30 AM

METHOD B: InfiniShell / SnapRotator

The next few steps will outline step by step how an InfiniBox storage admin would create immutable snapshots on a series of primary volumes utilizing the InfiniShell commands via InfiniBox SnapRotator toolset.

SnapRotator is a tool which runs externally to the InfiniBox on a hardened Linux OS to instrument large scale snapshot creation and management. SnapRotator can be configured to authenticate to the InfiniBox API and issue snapshot creation to all volumes or optionally leverage wildcards to select a series of volumes. SnapRotator can also be configured to integrate with VMWare to issue snapshots of VMWare Datastores informing the VMWare ESX environment to quiesce I/O to ensure there is a crash consistent snapshot of the VMGuests resident within the datastore.

OPTION 1:

Below is the native InfiniShell Command which can be issued to create an immutable snapshot. In this example we will again create an immutable snapshot of prod-vol02, however this lock expiry will be for only 3 days. InfiniShell is accessed via the InfiniBox GUI menu option in the upper right corner of the browser depicted by 3 horizontal bars. If you wish to use InfiniShell commands from an external host you can download the InfiniShell Package Set per the desired Operating System from support.infinidat.com.

Command Syntax:

```
vol.snap.create vol=prod-vol02 name=prod-vol02-snapshot-2 lock_duration=3DAYS
```

Command Validation Warning:

```
Setting / extending lock expiry for a snapshot will prevent any user from deleting it and releasing its capacity until 4/18/2022 14:23:56 UTC.
Are you sure? [y/n]
```

**Note if you wish to avoid validation for scripting purposes append ' -y ' to the command syntax*

Command Completion Syntax:

```
Volume snapshot "prod-vol02-snapshot-2" created. locked until 'date & time'
```

OPTION 2:

Below is SnapRotator configuration file user guide URL from the Infinidat support site

https://support.infinidat.com/hc/en-us/articles/360003909237-InfiniBox-SnapRotator-User-Guide

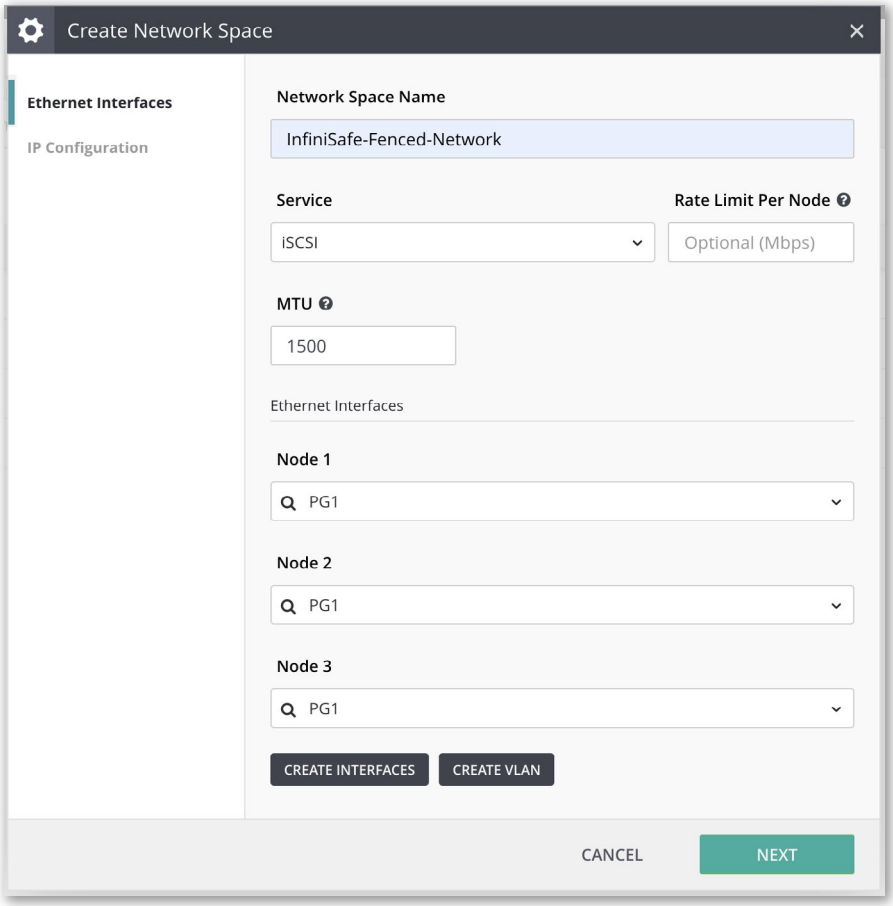
PHASE 2: Validation - Fenced Forensic iSCSI Network

METHOD A: InfiniBox GUI

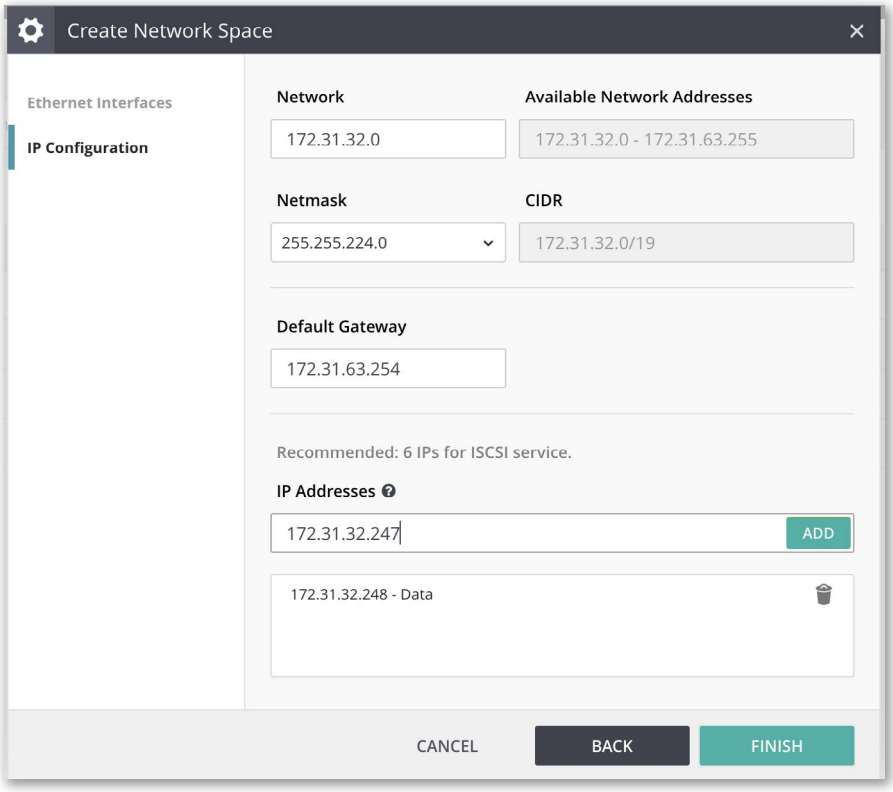
The next few steps will outline step by step how an InfiniBox storage admin would manually create an isolated fenced iSCSI network space utilizing the InfiniBox GUI.

Step 1:
Select the InfiniBox "Settings" pane on the bottom left of the InfiniBox GUI Dashboard. Next, navigate to the "Network Spaces" tab and select the "Create" button. Name the network space, choose iSCSI as the service and select the interfaces to be used on each of the 3 InfiniBox nodes. (InfiniBox Nodes are the 3 active/active/active storage controllers).

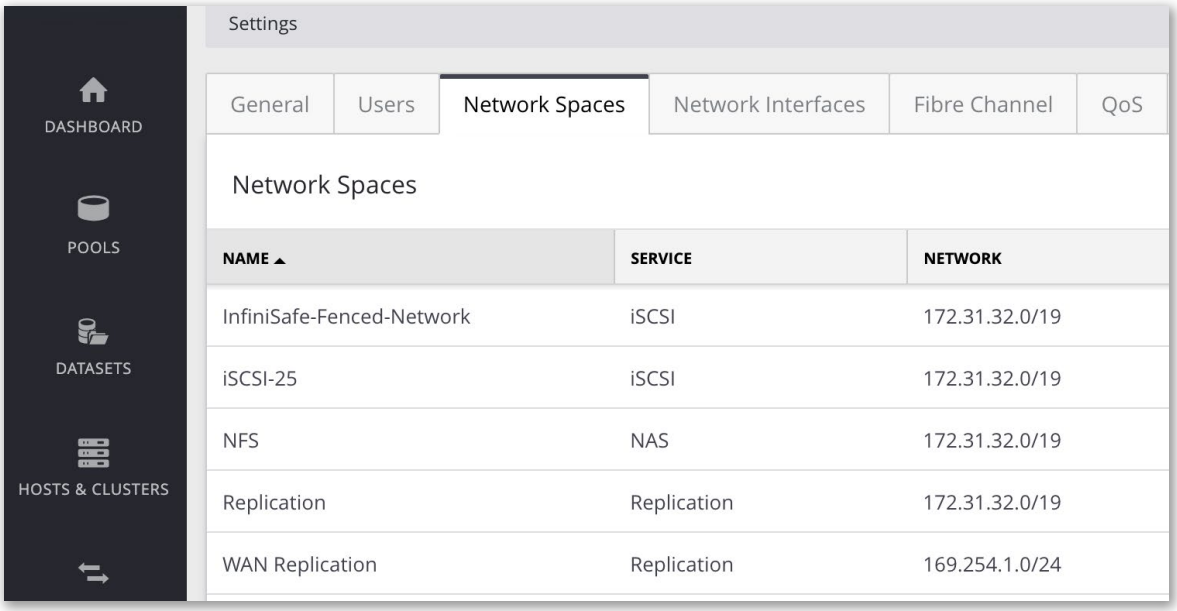
Optionally you can assign a VLAN tag if this iSCSI Network Space resides on a shared network by selecting the "Create VLAN" button during the initial network space creation.



Step 2:
Assign your desired IP addresses to the iSCSI Network Space. In this example we captured 2 of the recommended 6 addresses during the configuration assignment.



Upon completion the InfiniBox now has an isolated iSCSI network service called “InfiniSafe-Fenced-Network”



METHOD B: InfiniShell

The next few steps will outline step by step how an InfiniBox storage admin would create an isolated fenced iSCSI network space utilizing the InfiniShell, the syntax would be the same if issued from InfiniCLI on an external host.

Step 1:
Create iSCSI Network Space

Command Syntax:

config.net_space.create name=InfiniSafe-Fenced-Network service=ISCSI network=192.168.1.0/24 default_gateway=192.168.1.1

Command Completion Syntax:

Network Space “InfiniSafe-Fenced-Network” created

Step 2:
Assign IP Addresses to Network Space “InfiniSafe-Fenced-Network”

Command Syntax:

config.net_space.ip.create net_space=InfiniSafe-Fenced-Network ip_address=192.168.1.100,192.168.1.101,192.168.1.103,192.168.1.104,192.168.1.105,192.168.1.106

Once the use of the clean room is complete, the InfiniBox storage admin can quickly tear down the Network Space “InfiniSafe-Fenced-Network” with the following commands.

Step 1:
Disable “InfiniSafe-Fenced-Network” IP Addresses

Command Syntax:

config.net_space.ip.disable net_space=InfiniSafe-Fenced-Network ip_address=192.168.1.100,192.168.1.101,192.168.1.103,192.168.1.104,192.168.1.105,192.168.1.106

Command Validation:

Are you sure? [y/n] y

Command Completion Syntax:

IP addresses disabled in network space "InfiniSafe-Fenced-Network": 192.168.1.100,192.168.1.101,192.168.1.103,192.168.1.104,192.168.1.105,192.168.1.106

Step 2:

Delete IP Addresses in "InfiniSafe-Fenced-Nework" Network Space

Command Syntax:

config.net_space.ip.delete net_space=InfiniSafe-Fenced-Network ip_address=192.168.1.100,192.168.1.101,192.168.1.103,192.168.1.104,192.168.1.105,192.168.1.106

Command Validation:

Are you sure? [y/n] y

Command Completion Syntax:

IP addresses disabled in network space "InfiniSafe-Fenced-Network": 192.168.1.100,192.168.1.101,192.168.1.103,192.168.1.104,192.168.1.105,192.168.1.106

Step 3:

Delete "InfiniSafe-Fenced-Network" Network Space

Command Syntax:

config.net_space.delete net_space=InfiniSafe-Fenced-Network

Command Completion Syntax:

Network space "InfiniSafe-Fenced-Network" deleted

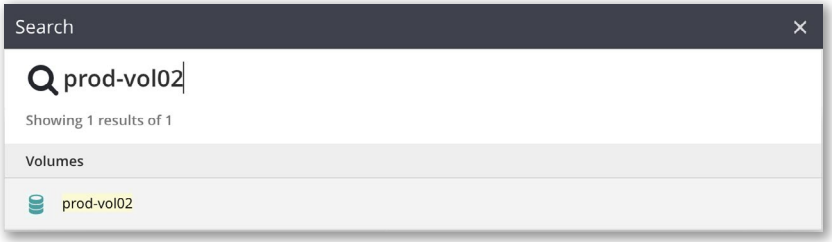
PHASE 3: Replicas - Remote Logical Air Gap

METHOD A: InfiniBox GUI

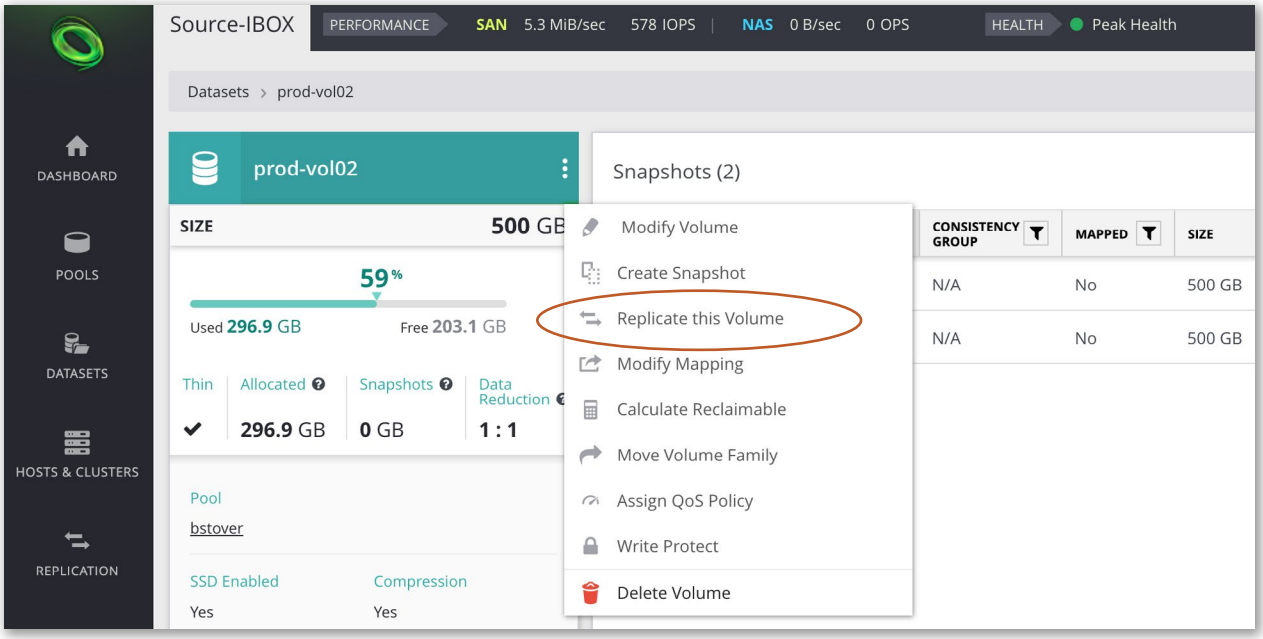
The next few steps will outline step by step how an InfiniBox storage admin using InfiniBox GUI would create a Remote logical Air Gap by replicating "prod-vol02" from "Source-IBOX" to "target ibox" and then creating an immutable snapshot on the "prod-vol02-target" located on the "Target-IBOX".

Step 1:

Locate the desired volume, this can be accomplished by using the general search window in the upper right hand side of the main InfiniBox dashboard. In this example we will search and select "prod-vol02"

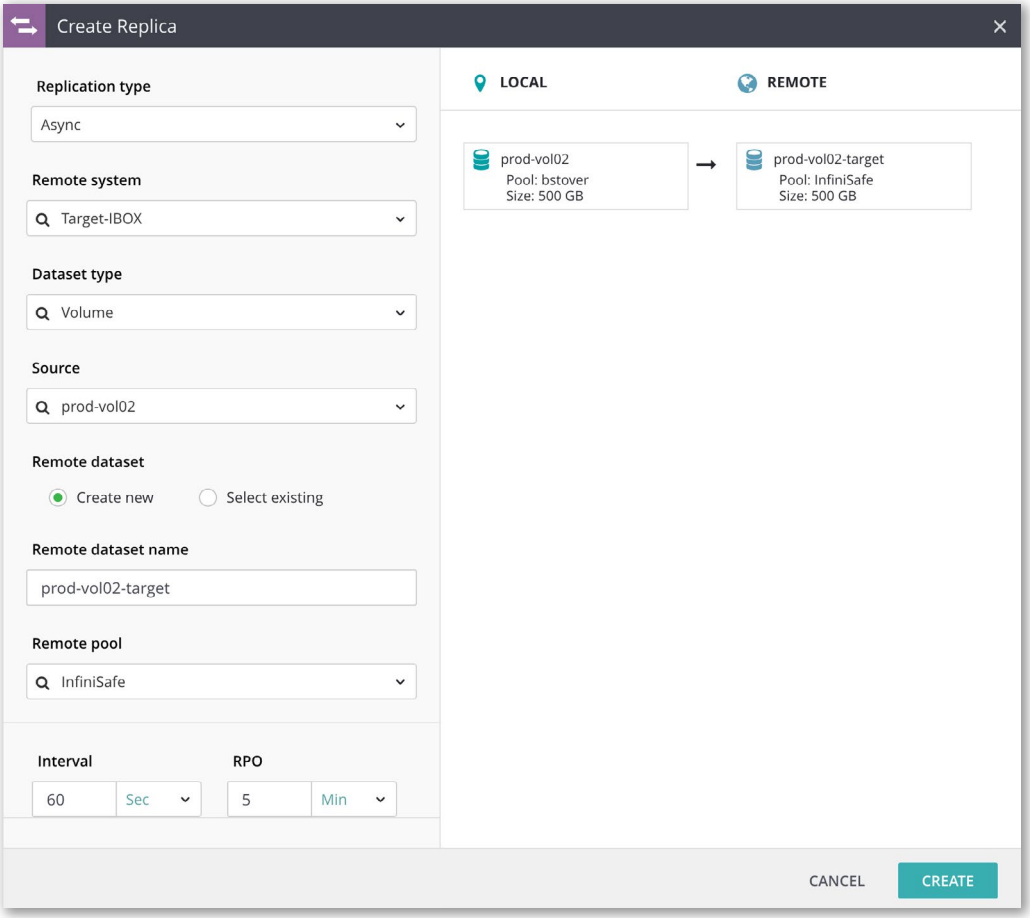


Step 2:
Right click on the prod-vol02 object using the 3 vertical dots to select "Replicate this volume"



Step 3:
Create Replica with replication type of Async to Remote System "Target-IBOX". The Create Replica window will automatically append "-target" to the source volume name "prod-vol02-target". Select the desired Remote pool on the "Target-IBOX" in this example the InfiniBox storage admin chose pool "InfiniSafe". The default Interval and default RPO were left intact for this example.

Once the prod-vol02 replica status is in an operational state of "Active" the InfiniBox storage admin is able to issue the create snapshot on the Target-IBOX system "prod-vol02-target" at the desired frequency and lock intervals described earlier in this document on the Source-IBOX prod-vol02.



METHOD B: InfiniShell

The following command will create an async replica volume with the name "prod-vol01-target" using a replication interval of 60 seconds and a recovery point objective of 5 minutes.

Command Syntax:

```
replica.create source=prod-vol01 replication_type=ASYNC system=Target-IBOX new_target_name=prod-vol01-target
remote_pool=InfiniSafe interval="00:60" rpo="00:05:00"
```

Command Completion Syntax:

```
Replica for volume "prod-vol01" created
```

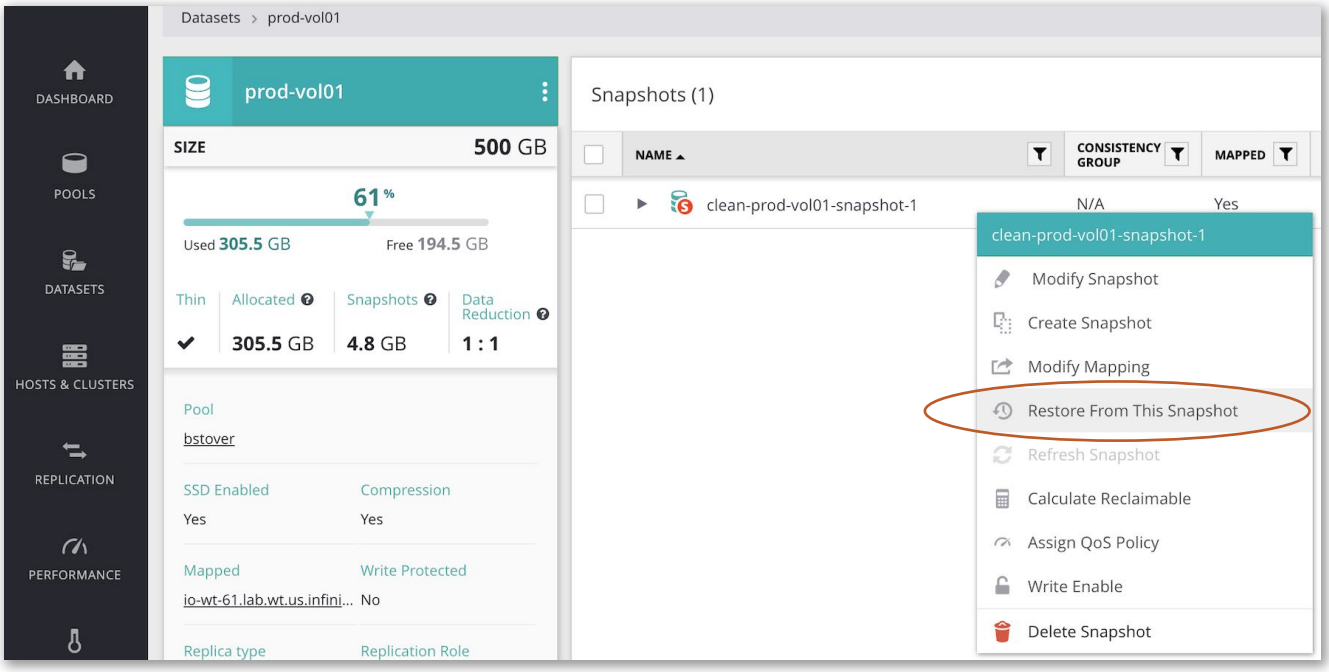
PHASE 4: Recovery - Immutable Snapshot Restoration

METHOD A: InfiniBox GUI

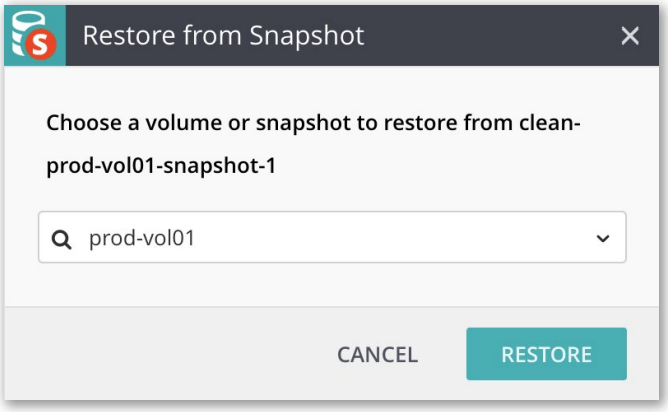
The next few steps will outline step by step how an InfiniBox storage admin using InfiniBox GUI would recover "prod-vol01" from immutable snapshot "clean-prod-vol01-snapshot-1".

Step 1:
Locate and select "prod-vol01" using the InfiniBox search icon, which is an icon of a magnifying glass in the upper right corner of the InfiniBox dashboard.

Step 2:
Locate "clean-prod-vol01-snapshot-1" from the list of available snapshots linked to prod-vol01. Right click on the "clean-prod-vol01-snapshot-1" and select "Restore From This Snapshot".



Step 3:
In the Restore from Snapshot dropdown box select "prod-vol01", note that if this were a multi series of linked snapshots an InfiniBox storage admin could opt to restore a non locked previous snapshot with an immutable snapshot further down the hierarchy of snapshots. In this example the only validated object able to be restored is the primary volume "prod-vol01" * .



* Additional consideration should be given to how the Operating System and or application will behave during the InfiniBox restore process. At a minimum the application should be taken offline. During an InfiniBox restore process the volume id or host OS UUID will remain the same, therefore the host operating system should not be impacted. Using InfiniSnap restore the InfiniBox is simply updating the primary volume metadata elements to point to the prior immutable snapshot metadata pointers. This is an in memory process which takes virtually no time and the application can be restarted as soon as the host Operating system has rescanned the bus to ensure no access change has occurred on the host side. Also of note, the immutable snapshot used to restore a primary volume will remain locked even after restoring the primary volume.

METHOD B: InfiniShell

The following command will restore volume "prod-vol01" from immutable snapshot "clean-prod-vol01-snapshot-1"

Command Syntax:

```
vol.restore vol=prod-vol01 source=clean-prod-vol01-snapshot-1
```

Command Validation Syntax:

```
Restoring volume 'prod-vol01' will overwrite its data.
Are you sure? [y/n] y
```

Command Completion Syntax:

```
Volume "prod-vol01" restored from snapshot "clean-prod-vol01-snapshot-1"
```

Appendix B

The following outlines the Ansible Playbooks in logical order of operations for the InfiniSafe security elements defined in the InfiniSafe Reference Architecture. Consult your Infinidat Technical Sales Engineer if you encounter a step needing assistance.

PHASE 1: Protection - Immutable Snapshots

```
$ cat inventory
[forensics]
io-wt-35.sample.lab.com

$ ansible-playbook
–-ask_become_pass
    --inventory “inventory”
    --extra-vars “@../ibox_vars/iboxCICD.yaml”
    --vault-password-file ../vault_password.txt
    “infinisafe_demo_setup.yml

PLAY [localhost] ********************************************

TASK [Gathering Facts] ********************************************
ok: [localhost]

TASK [Create pool infinisafe] ********************************
changed: [localhost]

TASK [Create volume app_vol under pool infinisafe] *************
changed: [localhost]

TASK [Create and lock (1 minute) snapshot app_snap from volume app_vol] ***
changed: [localhost]

PLAY RECAP ********************************************************
localhost              : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

PHASE 2: Validation and Recovery - Fenced Forensic iSCSI Network

```
$ ansible-playbook
   –-ask_become_pass
   --inventory “inventory”
   --extra-vars “@../ibox_vars/iboxCICD.yaml”
   --vault-password-file ../vault_password.txt
   “infinisafe_demo_runtest.yml”
BECOME password:

PLAY[forensics] *********************************************
```

```
TASK [Create ISCSI_SERVICE network space named InfiniSafe-Fenced-Network] ***
changed: [io-wt-35.sample.lab.com]

TASK [Create forensic host forensic-validation-host] ***********
changed: [io-wt-35.sample.lab.com]

TASK [Map snapshot app_snap to host forensic-validation-host] **
changed: [io-wt-35.sample.lab.com]

TASK [Add port to host forensic-validation-host] ***************
changed: [io-wt-35.sample.lab.com]

TASK [Connect forensics host forensic-validation-host to Infinibox ibox1521] ********************
changed: [io-wt-35.sample.lab.com]

TASK [Forensically test snapshot app_snap is clean using host forensic-validation-host] *********
changed: [io-wt-35.sample.lab.com]

TASK [debug] *******************************************************************************
ok: [io-wt-35.sample.lab.com] => {
   "msg": "Snapshot app_snap PASSED testing"
}

TASK [debug] *******************************************************************************
skipping: [io-wt-35.sample.lab.com]

TASK [Restoring volume app_vol from known clean snapshot app_snap] *****************************
changed: [io-wt-35.sample.lab.com]

PLAY RECAP *********************************************************************************
io-wt-35.sample.lab.com : ok=8    changed=7    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

PHASE 3: Forensic environment teardown

```
$ ansible-playbook
   --ask_become_pass
   --inventory "inventory"
   --extra-vars "@../ibox_vars/iboxCICD.yaml"
   --vault-password-file ../vault_password.txt
   "infinisafe_demo_teardown.yml"
BECOME password:

PLAY [forensics] ***********************************************

TASK [Unmap snapshot app_snap from host forensic-validation-host] ***
changed: [io-wt-35.sample.lab.com]
```

```
TASK [Remove port from host forensic-validation-host] **********
changed: [io-wt-35.sample.lab.com]

TASK [Disconnect forensics host forensic-validation-host from Infinibox ibox1521] ***
changed: [io-wt-35.sample.lab.com]

TASK [Remove network space named InfiniSafe-Fenced-Network] ****
changed: [io-wt-35.sample.lab.com]

TASK [Remove snapshot app_snap created from volume app_vol] ****
changed: [io-wt-35.sample.lab.com]

TASK [Remove volume app_vol under pool infinisafe] *************
changed: [io-wt-35.sample.lab.com]

TASK [Remove pool infinisafe] **********************************
changed: [io-wt-35.sample.lab.com]

TASK [Remove forensic host forensic-validation-host] ***********
changed: [io-wt-35.sample.lab.com]

PLAY RECAP *****************************************************
io-wt-35.sample.lab.com : ok=8    changed=8    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Playbook - infinisafe_demo_setup.yml

```
---
- hosts: localhost
  gather_facts: True  # Required for ansible_date_time
  collections:
    - infinidat.infinibox
  vars:
    network_space: InfiniSafe-Fenced-Network # iSCSI
    service: ISCSI_SERVICE
    pool: infinisafe
    volume: app_vol
    snap: app_snap
    host: forensic-validation-host
    host_iqn: iqn.1993-08.org.debian:01:62ebda3b76cc # io-wt-35
  tasks:

  - name: Create pool {{ pool }}
    infini_pool:
      name: "{{ pool }}"
      size: 1TB
      vsize: 1TB
      state: present
```

```
    user: "{{ user }}"
    password: "{{ password }}"
    system: "{{ system }}"

 - name: Create volume {{ volume }} under pool {{ pool }}
   infini_vol:
    name: "{{ volume }}"
    size: 1GB
    pool: "{{ pool }}"
    state: present

    user: "{{ user }}"
    password: "{{ password }}"
    system: "{{ system }}"

 - name: Create and lock (1 minute) snapshot {{ snap }} from volume {{ volume }}
   infini_vol:
    name: "{{ snap }}"
    state: present
    volume_type: snapshot
    parent_volume_name: "{{ volume }}"
    snapshot_lock_expires_at: "{{ ansible_date_time.iso8601_micro | to_datetime(fmt) | infinidat.infinibox.delta_
time(minutes=1) }}"

    user: "{{ user }}"
    password: "{{ password }}"
    system: "{{ system }}"
   vars:
    fmt: "%Y-%m-%dT%H:%M:%S.%fZ"
```

Playbook - infinisafe_demo_runtest.yml

```
---
### Localhost
- hosts: forensics
  gather_facts: False
  collections:
    - infinidat.infinibox
  vars:
    network_space: InfiniSafe-Fenced-Network # iSCSI
    service: ISCSI_SERVICE
    pool: infinisafe
    volume: app_vol
    snap: app_snap
    host: forensic-validation-host
    host_iqn: iqn.1993-08.org.debian:01:62ebda3b76cc # io-wt-35
```

```
  ibox_portal: 10.10.10.145
  ibox: ibox1521
  ibox_iqn: iqn.2009-11.com.infinidat:storage:infinibox-sn-1521
tasks:

- name: Create {{ service }} network space named {{ network_space }}
  infini_network_space:
    name: "{{ network_space }}"
    state: present
    service: "{{ service }}"
    interfaces:
    - 1680
    - 1679
    - 1678
    netmask: 19
    network: 10.10.10.0
    default_gateway: 10.10.10.254
    # rate_limit: 8
    # mtu: 1280
    ips:
    - 10.10.10.145
    - 10.10.10.146
    - 10.10.10.147
    - 10.10.10.148
    - 10.10.10.149
    - 10.10.10.150

    user: "{{ user }}"
    password: "{{ password }}"
    system: "{{ system }}"
  delegate_to: localhost

    user: "{{ user }}"
    password: "{{ password }}"
    system: "{{ system }}"
  delegate_to: localhost

- name: Create forensic host {{ host }}
  infini_host:
    name: "{{ host }}"
    state: present

    user: "{{ user }}"
    password: "{{ password }}"
    system: "{{ system }}"
  delegate_to: localhost
```

```
  - name: Map snapshot {{ snap }} to host {{ host }}
    infini_map:
      host: "{{ host }}"
      volume: "{{ snap }}"
      state: present

      user: "{{ user }}"
      password: "{{ password }}"
      system: "{{ system }}"
    delegate_to: localhost

  - name: Add port to host {{ host }}
    infini_port:
      host: "{{ host }}"
      iqns: "{{ host_iqn }}"
      state: present

      user: "{{ user }}"
      password: "{{ password }}"
      system: "{{ system }}"
    delegate_to: localhost

### Forensics Host
  - name: Connect forensics host {{ host }} to Infinibox {{ ibox }}
    shell: |
      iscsiadm --mode discoverydb --type sendtargets --portal {{ ibox_portal }} --discover
      iscsiadm --mode node --targetname={{ ibox_iqn }} --op update \
        --name=node.session.auth.username --value={{ user }}
      iscsiadm --mode discovery --type sendtargets --portal {{ ibox_portal }} --op show
      iscsiadm --mode node --targetname {{ ibox_iqn }} --portal {{ ibox_portal }} --login
      rescan-scsi-bus.sh
    become: yes

  # Run forensic tests on snapshot {{ snap }}
  - name: Forensically test snapshot {{ snap }} is clean using host {{ host }}
    shell: |
      true
    register: is_snapshot_clean

### Localhost
  - debug:
      msg: Snapshot {{ snap }} PASSED testing
    when: is_snapshot_clean.rc == 0
    delegate_to: localhost

  - debug:
      msg: Snapshot {{ snap }} FAILED testing. Do not use this snapshot.
```

```
  when: is_snapshot_clean.rc != 0
  delegate_to: localhost

 - name: Restoring volume {{ volume }} from known clean snapshot {{ snap }}
  infini_vol:
    name: "{{ snap }}"
    state: present
    parent_volume_name: "{{ volume }}"
    volume_type: snapshot
    restore_volume_from_snapshot: True

    user: "{{ user }}"
    password: "{{ password }}"
    system: "{{ system }}"
  when: is_snapshot_clean.rc == 0
  delegate_to: localhost
```

Playbook - infinisafe_demo_teardown.yml

```
---
### Localhost
- hosts: forensics
 gather_facts: False
 collections:
  - infinidat.infinibox
 vars:
  network_space: InfiniSafe-Fenced-Network # iSCSI
  service: ISCSI_SERVICE
  pool: infinisafe
  volume: app_vol
  snap: app_snap
  host: forensic-validation-host
  host_iqn: iqn.1993-08.org.debian:01:62ebda3b76cc # io-wt-35
  ibox_portal: 10.10.10.145
  ibox: ibox1521
  ibox_iqn: iqn.2009-11.com.infinidat:storage:infinibox-sn-1521
  ibox_portals: 10.10.10.148 10.10.10.146 10.10.10.149 10.10.10.145 10.10.10.150 10.10.10.147
 tasks:

 - name: Unmap snapshot {{ snap }} from host {{ host }}
  infini_map:
    host: "{{ host }}"
    volume: "{{ snap }}"
    state: absent
```

```
    user: "{{ user }}"
    password: "{{ password }}"
    system: "{{ system }}"
  delegate_to: localhost

- name: Remove port from host {{ host }}
  infini_port:
    host: "{{ host }}"
    iqns: "{{ host_iqn }}"
    state: absent

    user: "{{ user }}"
    password: "{{ password }}"
    system: "{{ system }}"
  delegate_to: localhost

### Forensics Host
- name: Disconnect forensics host {{ host }} from Infinibox {{ ibox }}
  shell: |
    for i in {{ ibox_portals }}; do
      iscsiadm --mode node --target {{ ibox_iqn }} -p $i --logout
    done
    for i in {{ ibox_portals }}; do
      iscsiadm --mode discoverydb -t sendtargets -p $i -o delete --discover
    done
  become: yes

### Localhost
- name: Remove network space named {{ network_space }}
  infini_network_space:
    name: "{{ network_space }}"
    state: absent

    user: "{{ user }}"
    password: "{{ password }}"
    system: "{{ system }}"
  delegate_to: localhost

- name: Remove snapshot {{ snap }} created from volume {{ volume }}
  infini_vol:
    name: "{{ snap }}"
    state: absent

    user: "{{ user }}"
    password: "{{ password }}"
    system: "{{ system }}"
  delegate_to: localhost
```

```
- name: Remove volume {{ volume }} under pool {{ pool }}
  infini_vol:
    name: "{{ volume }}"
    pool: "{{ pool }}"
    state: absent

    user: "{{ user }}"
    password: "{{ password }}"
    system: "{{ system }}"
  delegate_to: localhost

- name: Remove pool {{ pool }}
  infini_pool:
    name: "{{ pool }}"
    state: absent

    user: "{{ user }}"
    password: "{{ password }}"
    system: "{{ system }}"
  delegate_to: localhost

- name: Remove forensic host {{ host }}
  infini_host:
    name: "{{ host }}"
    state: absent

    user: "{{ user }}"
    password: "{{ password }}"
    system: "{{ system }}"
  delegate_to: localhost
```