ROSE-storage

Summary

Packages required: `rose-storage`

ROSE - package adds additional enterprise data center functionality to RouterOS - for supporting disk monitoring, improved formatting, RAIDs, rsync, iSCSI ,NVMe over TCP, NFS and improved SMB. This functionality currently is supported on **arm, arm64, x86** and **tile** platforms.

```
/disk
```

General inteface properties

Property	Description
crypted-backend	Drive or device used together with type=crypted to make "dm_crypt" encrypted storage
encryption-key	
iscsi-address	
iscsi-export	
iscsi-iqn	
iscsi-port	
nfs-address	
nfs-export	
nfs-share	
nvme-tcp-address	

nvme-tcp-export	
nvme-tcp-host-name	
nvme-tcp-name	
nvme-tcp-password	
nvme-tcp-port	
nvme-tcp-server-allow-host-name	
nvme-tcp-server-password	
nvme-tcp-server-port	
raid-chunk-size	
raid-device-count	
raid-master	
raid-max-component-size	
raid-member-failed	
raid-role	
raid-type	
slot	
smb-address	
smb-encryption	
smb-export	
smb-password	
smb-share	
smb-user	
tmpfs-max-size	
type	

Partitions

GPT partitions are supported starting from RouterOS 7.8beta3

to add 500MB partition:

```
/disk
disk add type=partition parent=sata1 partition-size=500M
```

if next partition will be added it will automatically allocated in available space from start of drive.

partition can be also added with offset:

```
/disk
add type=partition parent=sata1 partition-size=500M partition-offset=10G
```

on partition overlap, RouterOS will return error.

RAID

RAID (Redundant Array of Independent Disks) technology allows storing data on multiple drives - improving data transfer performance, data protection or both by combining them into logical units.

RAID levels

RouterOS supports software RAID levels 0,1,4,5,6,linear and nested RAID.

RAID 0

All data is written evenly over all disks in this RAID, this configuration does not provide any fault tolerance but provides best performance.

RAID 1

Same data is written in all drives (data is mirrored), this configuration provides best fault tolerance, but performance wise write speeds will be equal to slowest disk used in array.

RAID 4

Block-level data is striped to a dedicated disk where parity bits are stored. Performance will be limited to a parity writing speed.

RAID 5

Block-level data is striped evenly over the available disks. Can be recovered from 1 disk failure.

RAID 6

Block-level data is striped evenly over the available disks. Can be recovered from 2 disk failures.

Linear

Data is appended over multiple disks combining them into single large disk. Provides no redundancy and is limited to single disk read/write speed.

Nested RAID

Combination of multiple RAID configurations into other RAID. For example RAID 10 (RAID 1+0) combines disk mirroring (RAID 1) and disk striping (RAID 0)

RAID configuration

In this example we will create RAID 6 with 10 disks
Disks has to be in same size or have to be set in same size partitions or use raid-max-component-size parameter to limit larger volume size to match other elements.

In theory for RAID performance optimization - correct stride and stripe-width should be used. These are dependent on the RAID "raid-chunk-size", filesystem block size, and the number of disks.

```
stride=raid-chunk-size/block_size
```

```
stripe_width=disks*stride
```

RouterOS does this automatically when formatting local RAID device.

Create RAID device:

```
/disk add type=raid raid-type=6 raid-device-count=10 slot=raid1
```

add disks into this raid:

```
/disk set pcie1-nvme1 raid-master=raid1 raid-role=0
/disk set pcie1-nvme2 raid-master=raid1 raid-role=1
/disk set pcie1-nvme3 raid-master=raid1 raid-role=2
/disk set pcie1-nvme4 raid-master=raid1 raid-role=3
/disk set pcie1-nvme5 raid-master=raid1 raid-role=4
/disk set pcie1-nvme6 raid-master=raid1 raid-role=5
/disk set pcie1-nvme7 raid-master=raid1 raid-role=6
/disk set pcie1-nvme8 raid-master=raid1 raid-role=7
/disk set pcie1-nvme9 raid-master=raid1 raid-role=8
/disk set pcie1-nvme10 raid-master=raid1 raid-role=9
```

where pcie1-nvme* your local disk slot name

ⓘ Setting "raid-role" manually is optional, but highly recommended. If device has never been in raids before then superblock is empty, and raid-role will be assumed automatically, if not there may be error regarding using same RAID role if that has been already taken.

Raid will now sync:

```
/disk print detail
...
20 bM      type=raid slot="raid1" slot-default="" parent=none device="md0" uuid="3b4d4ec9-e7413ae8-37e7e397-
9cd9152e"
           fs=ext4 model="RAID5 1-parity-disk" size=8 641 770 946 560 free=8 572 463 624 192 raid-type=5
           raid-device-count=10 raid-max-component-size=none raid-chunk-size=1M raid-master=none
           raid-state="clean, resync =  1.8% (17498368/937692160) finish=45.2min speed=339148K/sec"
           nvme-tcp-export=no iscsi-export=no nfs-export=no smb-export=no
```

iSCSI

iSCSI allows accessing storage over an IP-based network. On initiator iSCSI device will appear as block device. RouterOS supports both target and initiator modes.

Target (Host) configuration:

```
/disk
set pcie1-nvme1 iscsi-export=yes
```

Initiator (client):

```
/disk
add type=iscsi iscsi-address=192.168.1.1 iscsi-iqn=pcie1-nvme1
```

iscsi-iqn needs to match slot name on target device, iscsi-address is target address.

NFS

NFS allows sharing local directories over network. RouterOS currently supports NFS v4 only mode.

Host configuration:

```
/disk
set pcie1-nvme1 nfs-export=yes
```

Initiator (client):
RouterOS

```
/disk
add type=nfs nfs-address=192.168.1.1
```

Linux:

```
mkdir /mnt/files
mount -t nfs 192.168.1.1:/ /mnt/files
```

SMB

SMB is popular file sharing protocol. ROSE package currently supports SMB2.1 SMB3.0, SMB3.1.1 dialects (SMB1 is not supported due to security vulnerabilities)

RouterOS also supports older SMB without ROSE package - SMB with legacy protocol support.

Host configuration:

```
/disk
set pcie1-nvme1 smb-export=yes
```

Initiator (client):

```
/disk
add type=smb smb-address=192.168.1.1 smb-share=pcie1-nvme1
```

smb-share needs to match slot name on target device, smb-address is target address.

NVMe over TCP

nvme-tcp allows accessing storage over network as NVMe block device on initiator side. On target side this device can be hdd/ssd/nvme or even raid array.

Target (Host) configuration:

```
/disk
set pcie1-nvme2 nvme-tcp-export=yes nvme-tcp-port=4420
```

Initiator (client):
RouterOS

````
/disk
add type=nvme-tcp nvme-tcp-address=192.168.1.1 nvme-tcp-name=pcie1-nvme1
````

nvme-tcp-name needs to match slot name on target device.

Linux:

load kernel module

````
modprobe nvme_tcp
````

discover available nvme-tcp targets:

~~~
nvme discover -t tcp -a 192.168.1.1 -s 4420
```
Discovery Log Number of Records 1, Generation counter 2
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified, sq flow control disable supported
portid:  4420
trsvcid: 4420
subnqn:  pcie1-nvme1
traddr:  10.155.166.7
sectype: none
~~~

subnqn should match slot name and will be used as -n parameter:

````
nvme connect -t tcp -a 192.168.1.1 -s 4420 -n pcie1-nvme1
````

Block device now should be available:

~~~
ls /dev/nvme*
```
/dev/nvme0  /dev/nvme0n1  /dev/nvme-fabrics
~~~

To disconnect:

````
nvme disconnect -d /dev/nvme0
````

where /dev/nvme0 previously mounted device, or disconnect all:

Error rendering macro 'code': Invalid value specified for parameter '[Ljava.lang.Object;@6ab08a13'

````
nvme disconnect-all
````

RAMdisk

RAMdisk - allows using part of RAM as attached device (block device). If compared to tmpfs - this allows using RAM as part of raid, or any other configuration where device instead of folder is required.

```
/disk
disk add type=ramdisk ramdisk-size=500M
```

RAMdisk will be cleared on reboot or power loss

Data encryption

Currently RouterOS supports SED (Self-encrypting drives) and dm_crypt drive encryption.

Self-Encrypting Drives

For using SED - drives has to be Opal-compliant. Please consult drive manufacturers documentation to find out if particular drive supports this feature before buying drives.
RouterOS adds **o (supported inactive)** or **O (supported active)** flags for supported drives:

```
/disk print
Flags: B - BLOCK-DEVICE; M, F - FORMATTING; o - TCG-OPAL-SELF-ENCRYPTION-SUPPORTED
Columns: SLOT, MODEL, SERIAL, INTERFACE, SIZE, FREE, FS, RAID-MASTER
#     SLOT   MODEL                  SERIAL           INTERFACE                   SIZE              FREE  FS
RAID
0 BMo sata1  Samsung SSD 860 2.5in  S3Z9NX0N414510L  SATA 6.0 Gbps  1 000 204 886 016  983 351 111 680  ext4
none
1 BMo sata2  Samsung SSD 860        S5GENG0N307602J  SATA 6.0 Gbps  1 000 204 886 016  983 351 128 064  ext4
none
2 BMO sata3  Samsung SSD 860        S5GENG0N307604H  SATA 6.0 Gbps  1 000 204 886 016  983 351 128 064  ext4
none
3 BMO sata4  Samsung SSD 860 2.5in  S4CSNX0N838150B  SATA 6.0 Gbps  1 000 204 886 016  983 351 128 064  ext4
none
```

To set TCG-OPAL-SELF-ENCRYPTION:

```
/disk
disk set sata1 self-encryption-password=securepassword
```

to unset:

```
/disk
disk unset sata1 self-encryption-password
```

or

```
/disk
disk set sata1 !self-encryption-password
```

Block device encryption

...

File Sync

ROSE package also includes tool for file upload/download and sync.
To sync (push) local folder contents to other RouterOS device:

```
/file sync add local-path=pcie1-nvme1/myfolder/ remote-addrs=192.168.1.1 mode=upload user=admin password=""
remote-path=test/
```

where user/password - other device username and password. For interconnection Winbox port needs to be open.

or to pull files use:

```
/file sync add local-path=pcie1-nvme1/myfolder/ remote-addrs=192.168.1.1 mode=download user=admin password=""
remote-path=test/
```

Folder now will be in sync, all changes to files will be synced between device.

> ⚠ One folder should not have download and upload type to same target, to avoid undefined behavior.