Volume Arrays

One way to safeguard data in the Rapid Access Cloud is to have multiple copies of that data spanning multiple disks, which can be done at the software level using `mdadm` in Linux. The Block Storage volumes that get attached to instances can be assembled in any RAID configuration (raid6, raid5, raid4, raid0, raid1, raid10) depending on use-case.

Note: Creating multiple volumes will consume your overall Block Storage quota. We recommend only creating a volume array for situations where the data would be otherwise hard to restore.

Volume Affinity

When a volume is created in the Rapid Access Cloud, the Block Storage service provisions storage on an underlying storage server. If this storage server is ever taken offline due to hardware failure or maintenance, your volume will be unavailable. Similarly, if you create a volume array and the majority of volumes are all created on the same storage server, your entire array might become unavailable as well.

To protect against this, we recommend using Volume Affinity when creating a volume array. Volume Affinity is a way of creating volumes so that each volume is placed on a separate storage server.

You must use the "cinder" command line tool in order to use Volume Affinity.

1. Create the first volume. This example creates a 2GB volume called "vol_01":

```
$ cinder --os-volume-api-version 3 create --display-name vol_01 2
```

2. Note the UUID of the new "vol_01".
3. Create the second volume. This uses a scheduling "hint" that will place the volume on any storage server other than "vol_01":

```
$ cinder --os-volume-api-version 3 create --hint different_host=51824076-ff7e-469e-b373-b4b199fe32e9 --
display-name vol_02 2
```

4. Note the UUID of the new "vol_02".
5. Create a third volume that will be placed on any storage server other than where "vol_01" and "vol_02" reside:

```
$ cinder --os-volume-api-version 3 create --hint different_host=51824076-ff7e-469e-b373-b4b199fe32e9 --
hint different_host=b4a96d9d-b2c1-476f-a0cb-0c2f83cdb887 --display-name vol_03 2
```

6. You may continue repeating these steps, but be aware that Volume Affinity is limited to only two "different_host" options.

Once your volumes have been created, you can proceed with creating the volume array.

Creating the array

For this example assume a four volume RAID6 array with 4 100GB volumes.

1. Create an instance.
2. Create four 100 GB cinder volumes. You may use Volume Affinity to create the volumes (see the Volume Affinity section), though this is optional.
3. Attach the volumes to the instance.
    a. At this time, find out what device names each volume was given by the operating system:

```
$ lsblk
```

    (the volumes will most likely be given sequential names e.g. `/dev/sdc`, `/dev/sdd`, etc)
4. Edit `/etc/fstab` on the instance to define the swap volume by UUID instead of device name. The following one-line script will replace `/dev/sdb` with the UUID of the volume.

```
$ sudo sed -i "s/\/dev\/sdb/$(blkid | grep sdb | awk '{print $2}')/g" /etc/fstab
```

5. Partition and create a filesystem on each attached volume (devices `/dev/sdc` through `/dev/sdf`)
6. Create the array:

```
$ sudo mdadm --create /dev/md0 --level=6 --raid-devices=4 /dev/sdc1 /dev/sdd1 /dev/sde1 /dev/sdf1
```

7. Create a mdadm.conf file with the array UUID for recovery (see below for more).

```
$ sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/mdadm.conf
```

Assembling a previously created array

It is possible to reattach all the volumes and have the array assemble itself using the metadata on the volumes. This might need to be done if the instance needed to be relaunched or perhaps the data just moved to another instance.

This example is a continuation of the above example.

1. Attach the volumes to the array. If this is a new instance make sure you edit `/etc/fstab` as step 4 in Creating the array.
2. Check the status of the array:

```
$ cat /proc/mdstat
```

This should show something like this:

```
Personalities :
         md127 : inactive sdf1[3](S) sde1[2](S) sdd1[1](S) sdc1[0](S)
                          314373120 blocks super 1.2
         unused devices: <none>
```

3. Start the array:

```
$ sudo mdadm --manage /dev/md127 -R
```