OpenZFS

A Device by Any Other Name

Common Pitfalls in Device Naming for ZFS on Linux

●●●

Don Brady and Sara Hartse, Delphix

Topics

- Motivation
- Technical background
- Tools
- Practical examples
  - ESX, Azure, AWS, GCP
- Takeaways

Introduction

OpenZFS

There are a lot of different choices when it comes to identifying devices on Linux. It's important to choose the right ones to use with ZFS.

1. **zfs name** - ZFS needs to be able to uniquely and consistently identify devices so that it can reconstruct the same pool after exporting it

2. **display name** - You need to match devices in your VM/Cloud management software to the device they correspond to in your zpool, allowing you to add, remove and expand the devices you mean to.

Device References

- Create a pool

  -
    ```
    zpool create mypool sdb sdc sbc
    ```
- Import a pool

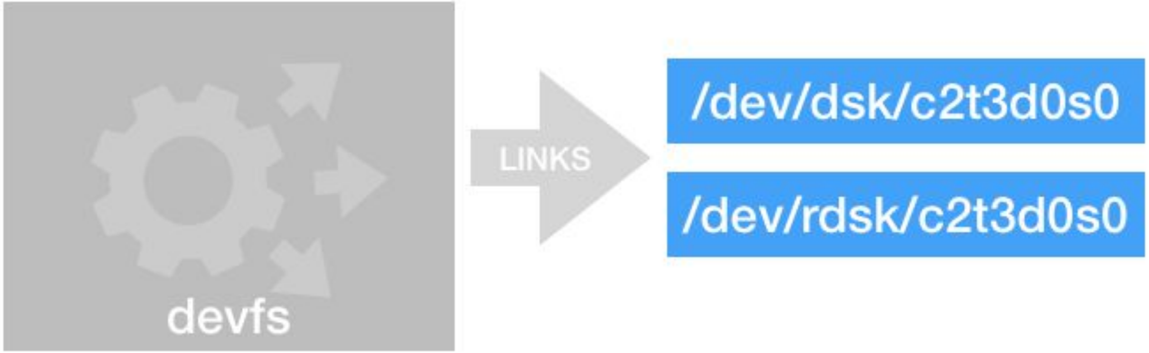
  -
    ```
    zpool import -d /dev/disk/by-id mypool
    ```
- Add a device

  -
    ```
    zpool add mypool /dev/disk/azure/scsi1/lun0
    ```
- Remove a device

  -
    ```
    zpool remove mypool xvdc
    ```
- Expand a device

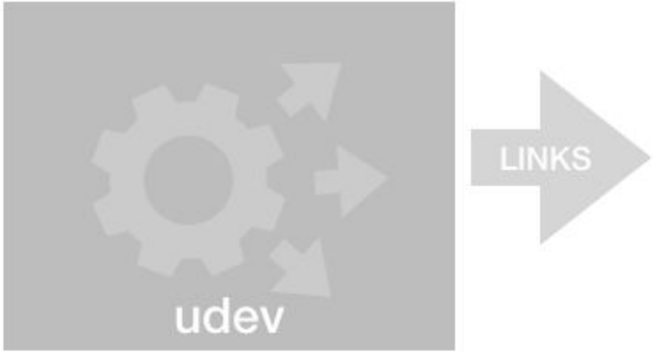
  -
    ```
    zpool online -e mypool
    /dev/disk/by-id/google-persistent-disk-1
    ```

illumos Device Links

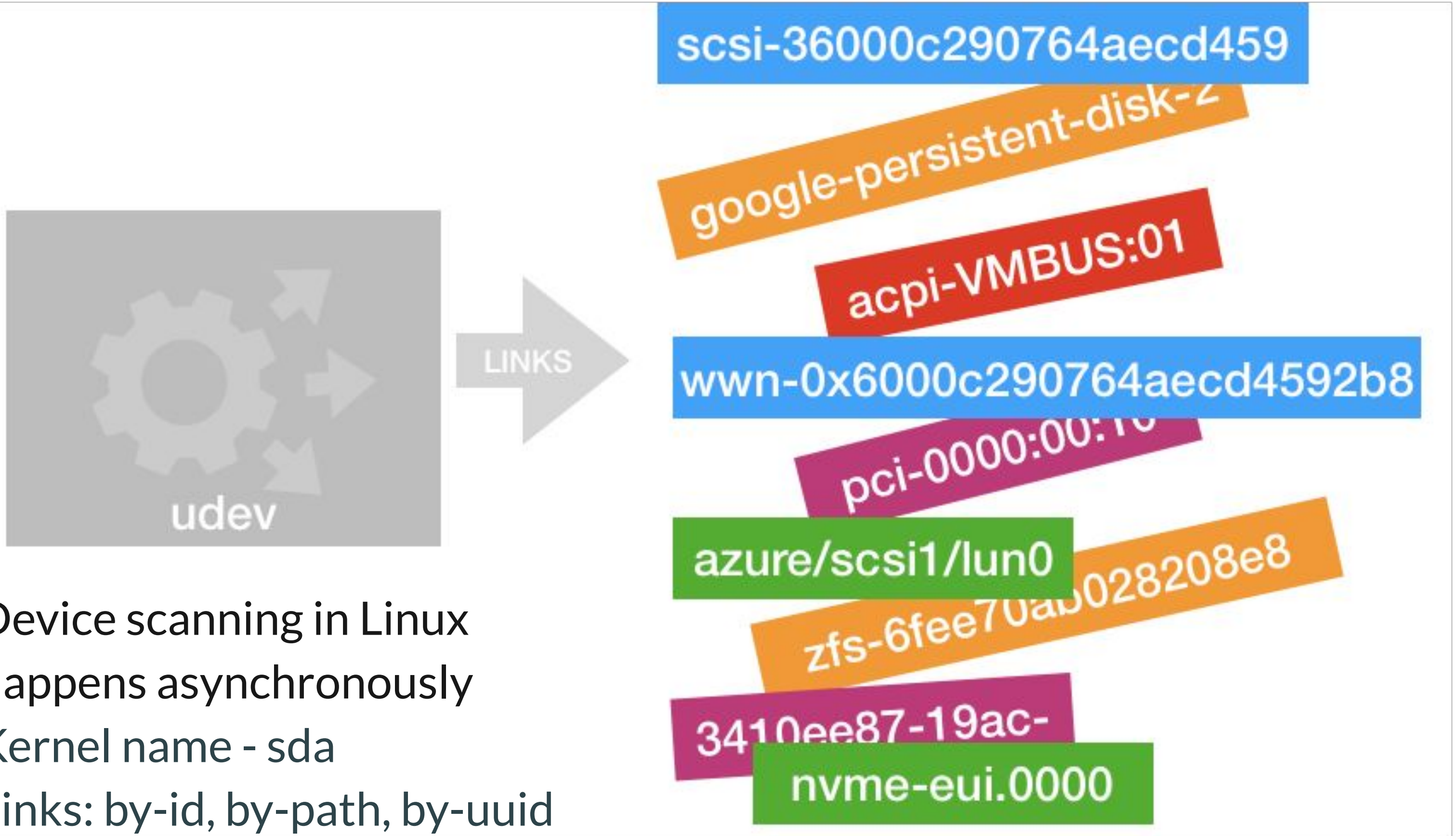


- Name - c2t3d0 - corresponds to the device’s location
- Devid - serial number - unchanging, unique to the device
- ZFS opens device with devid

Why are there so many choices?



Linux Device Links



- Device scanning in Linux happens asynchronously
- Kernel name - sda
- Links: by-id, by-path, by-uuid

sysfs & udev

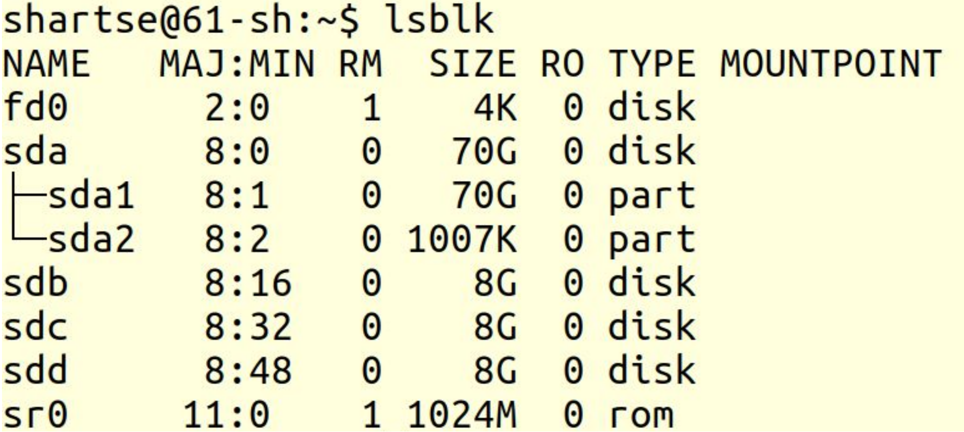
Sysfs

- a virtual file system managed by the Linux kernel
- exports information about devices from the kernel to userspace
- can also be used for controlling device configuration and state

Udev

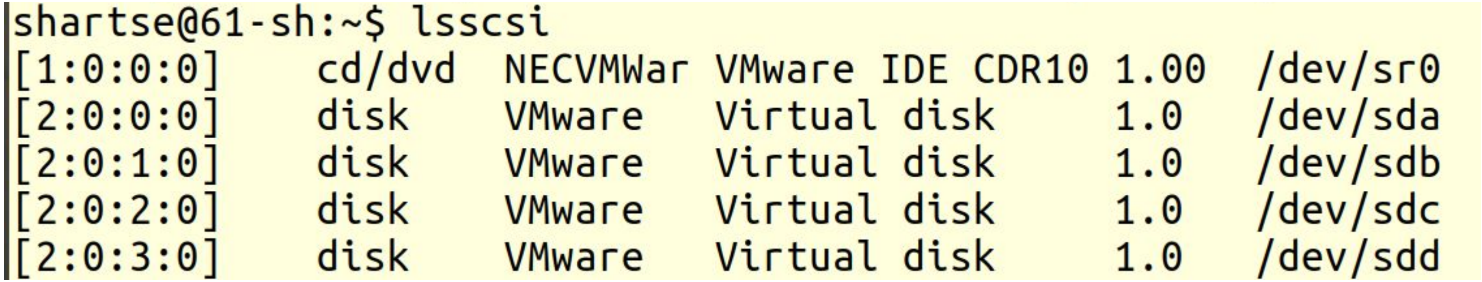
- udevd daemon runs in userspace
- notified when a kernel device is added or removed from the system
- automates the creation and removal of devices in '/dev' namespace
    - uses rules to specify what names are given to a device
    - allows for persistent/consistent naming schemes
    - consults sysfs to collect attributes and information used for naming

lsblk - list available block devices



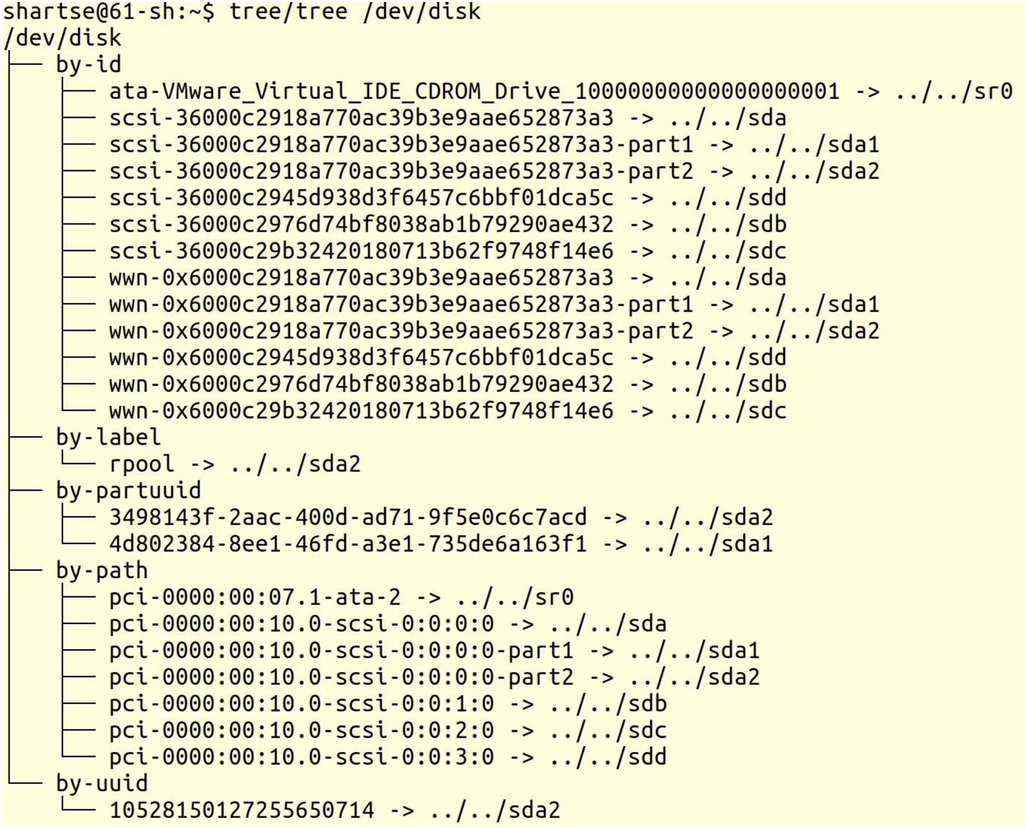
```
shartse@61-sh:~$ lsblk
NAME    MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
fd0       2:0    1     4K  0 disk
sda       8:0    0    70G  0 disk
├─sda1    8:1    0    70G  0 part
└─sda2    8:2    0 1007K  0 part
sdb       8:16   0     8G  0 disk
sdc       8:32   0     8G  0 disk
sdd       8:48   0     8G  0 disk
sr0      11:0    1 1024M  0 rom
```

lsscsi - limited to scsi devices



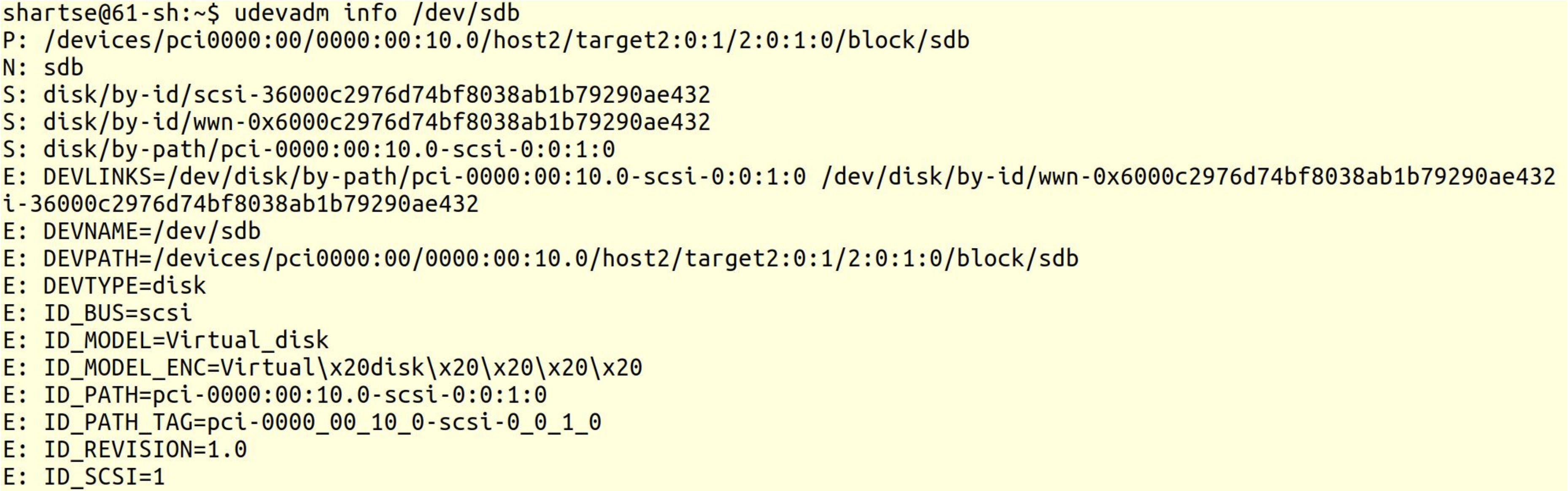
```
shartse@61-sh:~$ lsscsi
[1:0:0:0]    cd/dvd  NECVMWar VMware IDE CDR10 1.00  /dev/sr0
[2:0:0:0]    disk    VMware   Virtual disk     1.0   /dev/sda
[2:0:1:0]    disk    VMware   Virtual disk     1.0   /dev/sdb
[2:0:2:0]    disk    VMware   Virtual disk     1.0   /dev/sdc
[2:0:3:0]    disk    VMware   Virtual disk     1.0   /dev/sdd
```

`tree /dev/disk` - traverse links and show symlinks of all the device names



```
shartse@61-sh:~$ tree/tree /dev/disk
/dev/disk
├── by-id
│   ├── ata-VMware_Virtual_IDE_CDROM_Drive_10000000000000000001 -> ../../sr0
│   ├── scsi-36000c2918a770ac39b3e9aae652873a3 -> ../../sda
│   ├── scsi-36000c2918a770ac39b3e9aae652873a3-part1 -> ../../sda1
│   ├── scsi-36000c2918a770ac39b3e9aae652873a3-part2 -> ../../sda2
│   ├── scsi-36000c2945d938d3f6457c6bbf01dca5c -> ../../sdd
│   ├── scsi-36000c2976d74bf8038ab1b79290ae432 -> ../../sdb
│   ├── scsi-36000c29b32420180713b62f9748f14e6 -> ../../sdc
│   ├── wwn-0x6000c2918a770ac39b3e9aae652873a3 -> ../../sda
│   ├── wwn-0x6000c2918a770ac39b3e9aae652873a3-part1 -> ../../sda1
│   ├── wwn-0x6000c2918a770ac39b3e9aae652873a3-part2 -> ../../sda2
│   ├── wwn-0x6000c2945d938d3f6457c6bbf01dca5c -> ../../sdd
│   ├── wwn-0x6000c2976d74bf8038ab1b79290ae432 -> ../../sdb
│   └── wwn-0x6000c29b32420180713b62f9748f14e6 -> ../../sdc
├── by-label
│   └── rpool -> ../../sda2
├── by-partuuid
│   ├── 3498143f-2aac-400d-ad71-9f5e0c6c7acd -> ../../sda2
│   └── 4d802384-8ee1-46fd-a3e1-735de6a163f1 -> ../../sda1
├── by-path
│   ├── pci-0000:00:07.1-ata-2 -> ../../sr0
│   ├── pci-0000:00:10.0-scsi-0:0:0:0 -> ../../sda
│   ├── pci-0000:00:10.0-scsi-0:0:0:0-part1 -> ../../sda1
│   ├── pci-0000:00:10.0-scsi-0:0:0:0-part2 -> ../../sda2
│   ├── pci-0000:00:10.0-scsi-0:0:1:0 -> ../../sdb
│   ├── pci-0000:00:10.0-scsi-0:0:2:0 -> ../../sdc
│   └── pci-0000:00:10.0-scsi-0:0:3:0 -> ../../sdd
└── by-uuid
    └── 10528150127255650714 -> ../../sda2
```

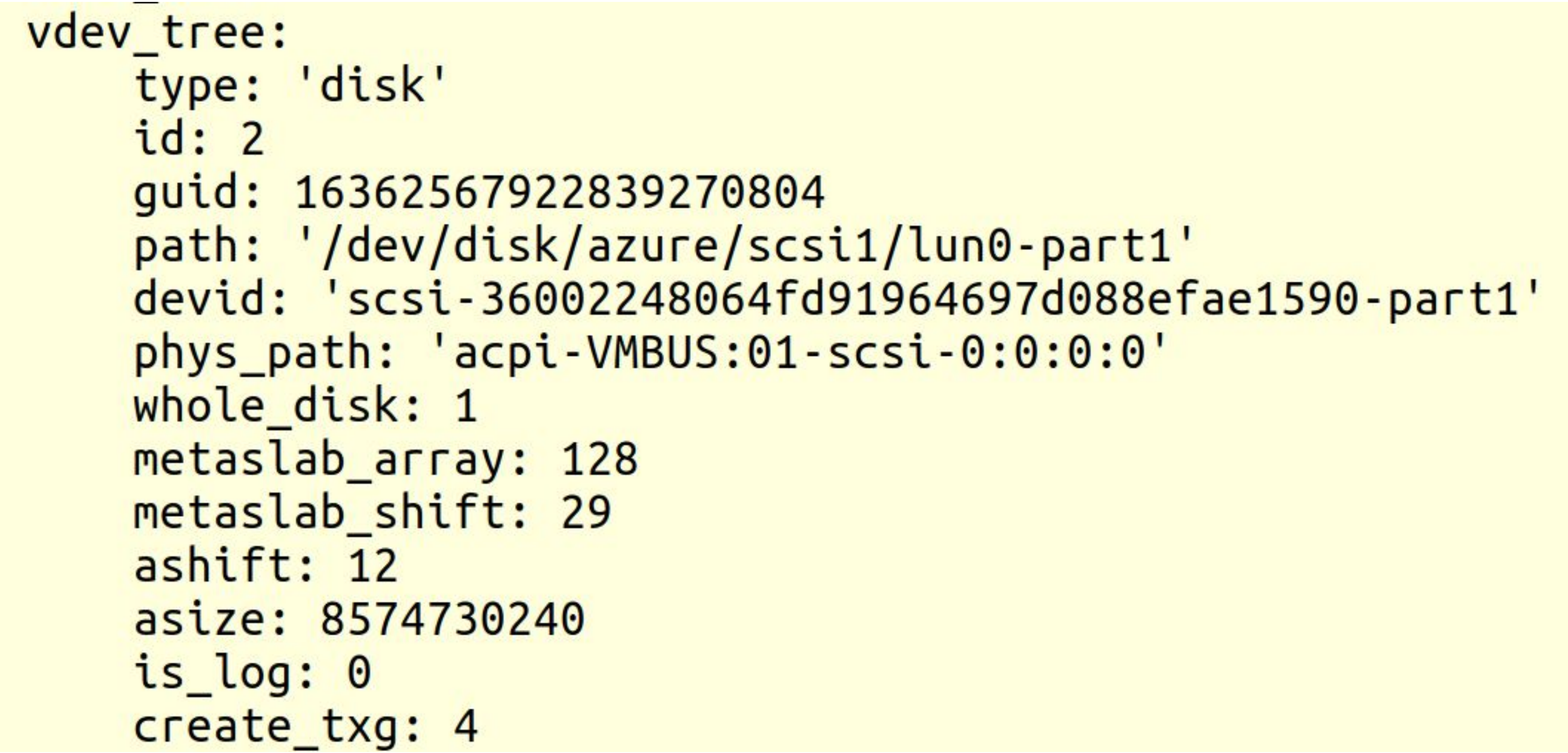
`udevadm info <devpath>` - Display all the different udev attributes available for a given device



```
shartse@61-sh:~$ udevadm info /dev/sdb
P: /devices/pci0000:00/0000:00:10.0/host2/target2:0:1/2:0:1:0/block/sdb
N: sdb
S: disk/by-id/scsi-36000c2976d74bf8038ab1b79290ae432
S: disk/by-id/wwn-0x6000c2976d74bf8038ab1b79290ae432
S: disk/by-path/pci-0000:00:10.0-scsi-0:0:1:0
E: DEVLINKS=/dev/disk/by-path/pci-0000:00:10.0-scsi-0:0:1:0 /dev/disk/by-id/wwn-0x6000c2976d74bf8038ab1b79290ae432
i-36000c2976d74bf8038ab1b79290ae432
E: DEVNAME=/dev/sdb
E: DEVPATH=/devices/pci0000:00/0000:00:10.0/host2/target2:0:1/2:0:1:0/block/sdb
E: DEVTYPE=disk
E: ID_BUS=scsi
E: ID_MODEL=Virtual_disk
E: ID_MODEL_ENC=Virtual\x20disk\x20\x20\x20\x20
E: ID_PATH=pci-0000:00:10.0-scsi-0:0:1:0
E: ID_PATH_TAG=pci-0000_00_10_0-scsi-0_0_1_0
E: ID_REVISION=1.0
E: ID_SCSI=1
```

`udevadm monitor <devpath>` - get a real-time log of udev events per device

`zdb -l <devpath>` - Dump device configuration as used by a vdev



```
vdev_tree:
    type: 'disk'
    id: 2
    guid: 16362567922839270804
    path: '/dev/disk/azure/scsi1/lun0-part1'
    devid: 'scsi-36002248064fd91964697d088efae1590-part1'
    phys_path: 'acpi-VMBUS:01-scsi-0:0:0:0'
    whole_disk: 1
    metaslab_array: 128
    metaslab_shift: 29
    ashift: 12
    asize: 8574730240
    is_log: 0
    create_txg: 4
```

ESX

- **Problem** Re-ordering devices supported in VMWare frontend, causing `/dev/sdN` device name to change. Can't find any other unique info.
- **Solution** With additional settings, we can enable device UUID links for the Linux OVA

```
DEVLINKS=
/dev/disk/by-id/scsi-36000c29c726057df4a5901c5068d533a
/dev/disk/by-path/pci-0000:00:10.0-scsi-0:0:2:0
/dev/disk/by-id/wwn-0x6000c29c726057df4a5901c5068d533a
```

- `/dev/disk/by-id` links work a zfs names (persistent) but are much more difficult to use display names.

AWS (Xen)

OpenZFS

- As far as we can tell, Xen /dev/disk entries are persistent
- There is no other unique identifier provided

```
$ udevadm info /dev/xvdb
P: /devices/vbd-51728/block/xvdb
N: xvdb
E: DEVNAME=/dev/xvdb
E: DEVPATH=/devices/vbd-51728/block/xvdb
```

- `xvdN` links work as zfs names (persistent) and as display names (short, match up with the AWS frontend).

Azure

- Has a /dev/by-id reference, but then we found that resizing the device changes the id
- Installed azure udev rules (comes with Azure Linux agent)

  ```
  DEVLINKS=
  /dev/disk/by-id/wwn-0x6002248064fd91964697d088efae1590
  /dev/disk/azure/scsi1/lun0
  /dev/disk/by-path/acpi-VMBUS:01-scsi-0:0:0:0
  /dev/disk/by-id/scsi-36002248064fd91964697d088efae1590
  ```

- `lunN` links well as a zfs names (persistent) and as display names (short, match up with the azure frontend).

GCP

- At this point, we started to see a pattern. If you have a /dev/by-id link, use it.
- /lib/udev/rules.d/65-gce-disk-naming.rules

```
DEVLINKS=
/dev/disk/by-id/scsi-0Google_PersistentDisk_persistent-disk-2
/dev/disk/by-id/google-persistent-disk-2
/dev/disk/by-path/pci-0000:00:03.0-scsi-0:0:3:0
```

- `/dev/disk/by-id` links work as zfs names (persistent) and, unlike for ESX, they're more human-intelligible

Conclusions

- Options for device naming are not consistent across different virtual platforms
- Take the time to understand which identifiers are available and most useful
- Test different device operations to see how identities behave
- Consider writing your own udev rules!

Questions?

Find us on OpenZFS Slack: @don.brady and @sara

Zpool events

```
Oct 29 2019 14:54:15.907495984 resource.fs.zfs.statechange
        version = 0x0
        class = "resource.fs.zfs.statechange"
        pool = "serenity"
        pool_guid = 0x732333af4cf5eab8
        pool_state = 0x0
        pool_context = 0x0
        vdev_guid = 0xc40694c1c50e9ef4
        vdev_state = "UNAVAIL" (0x4)
        vdev_path = "/dev/disk/by-id/scsi-350000394a8ca4fbc-part1"
        vdev_devid = "scsi-350000394a8ca4fbc-part1"
        vdev_physpath = "pci-0000:04:00.0-sas-phy0-lun-0"
        vdev_laststate = "ONLINE" (0x7)
        time = 0x5db8a6f7 0x36174a30
        eid = 0x12
```