CableOS Image Build

Try the deb live in a VM

- Boot with the debirf-live image
- Login with the default root:install account
- Create the /data directory
- Transfer the APOLLO_PLATFORM iso into /data
- Run the ostree-production list-isos to identify the file

```
0 debirf-live:~# ostree-production list-isos
I: List of ISO files under '/data' (product_code='' version=''):
    > APOLLO_PLATFORM-release-3.21.3.0-7+auto15.iso
I: Cleanup
```

- Apply the image to the disk

```
1 debirf-live:~# ostree-production -D /dev/sda from  /data/APOLLO_PLATFORM-
release-3.21.3.0-7+auto15.iso
I: ------------------------------------------------------------------------
---
I: Directly using '/data/APOLLO_PLATFORM-release-3.21.3.0-7+auto15.iso'
I: ------------------------------------------------------------------------
---
I: Handling product_code=APOLLO_PLATFORM version=3.21.3.0-7+auto15
Version: 3.21.3.0-7+auto15
I: Unexposing /tmp/APOLLO_PLATFORM-nsg-upgrade/iso-mounts/APOLLO_PLATFORM-
3.21.3.0-7+auto15
I: Exposing ISO: 3.21.3.0-7+auto15 (from /data/APOLLO_PLATFORM-release-
3.21.3.0-7+auto15.iso)
I: Exposed  ISO: 3.21.3.0-7+auto15 (under /tmp/APOLLO_PLATFORM-nsg-
upgrade/iso-mounts/APOLLO_PLATFORM-3.21.3.0-7+auto15)
I: Production to device '/dev/sda'

STEP-1 ==========================================================
Unmounting previous /mnt/passive

N: apollo-env(umount): skip -- already unmounted

STEP-2 ==========================================================
Partitioning disk and create file-systems

I: ------------------------------------------------------- REMOVE_ALL
I: ------------------------------------------------------- WIPE PREVIOUS
FILESYSTEM
I: ------------------------------------------------------- CREATE
PARTITIONS (msdos)
```

```
Information: You may need to update /etc/fstab.

Information: You may need to update /etc/fstab.

Information: You may need to update /etc/fstab.

Information: You may need to update /etc/fstab.

Information: You may need to update /etc/fstab.

Information: You may need to update /etc/fstab.

I: ----------------------------------------------------------- CREATE_ALL
(LVM)
  Physical volume "/dev/sda3" successfully created.
  Volume group "cos-slice-vg" successfully created
  Logical volume "root1" created.
  Logical volume "root2" created.
  Logical volume "common" created.
  Logical volume "confd" created.
  Logical volume "docker" created.
I: ----------------------------------------------------------- CREATE_ALL
(FILE-SYSTEMS)
mkfs.fat 4.2 (2021-01-31)
mke2fs 1.46.2 (28-Feb-2021)
/dev/sda2 contains a ext3 file system labelled 'GRUBPC'
    created on Tue Apr 23 20:31:14 2024
Creating filesystem with 488448 1k blocks and 122400 inodes
Filesystem UUID: e7c244c4-99c0-416c-9a20-cf90ab81281e
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

mke2fs 1.46.2 (28-Feb-2021)
Creating filesystem with 434176 4k blocks and 108640 inodes
Filesystem UUID: d4114381-5bb2-4289-b947-8d96a9ddadce
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

mke2fs 1.46.2 (28-Feb-2021)
Creating filesystem with 434176 4k blocks and 108640 inodes
Filesystem UUID: 1815dd30-6d1c-4633-b39d-6836a185263c
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912
```

```
Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

mke2fs 1.46.2 (28-Feb-2021)
Creating filesystem with 315392 1k blocks and 78936 inodes
Filesystem UUID: d88c0936-5617-4eb5-b91f-cacbf818c2e8
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729, 204801, 221185

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

mke2fs 1.46.2 (28-Feb-2021)
Creating filesystem with 2416640 4k blocks and 605024 inodes
Filesystem UUID: fc81f783-53ef-4e41-b005-1d2c15999940
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

mke2fs 1.46.2 (28-Feb-2021)
Creating filesystem with 585728 4k blocks and 146592 inodes
Filesystem UUID: fc97fc34-af12-4a41-97ed-73234d8c34da
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

I: ------------------------------------------------------------ READY
  LV     VG           Attr       LSize   Pool Origin Data%  Meta%  Move Log
Cpy%Sync Convert
  common cos-slice-vg -wi-a----- 308.00m
  confd  cos-slice-vg -wi-a-----   2.23g
  docker cos-slice-vg -wi-a-----  <9.22g
  root1  cos-slice-vg -wi-a-----  <1.66g
  root2  cos-slice-vg -wi-a-----  <1.66g

STEP-3 ============================================================
ROOTFS1 -- Mounting

I: Found disks:
/dev/disk/by-id/ata-VMware_Virtual_IDE_CDROM_Drive_00000000000000000001
/dev/disk/by-id/dm-name-cos--slice--vg-common
/dev/disk/by-id/dm-name-cos--slice--vg-confd
```

```
/dev/disk/by-id/dm-name-cos--slice--vg-docker
/dev/disk/by-id/dm-name-cos--slice--vg-root1
/dev/disk/by-id/dm-name-cos--slice--vg-root2
/dev/disk/by-id/dm-uuid-LVM-
bCUwiDLYkH0OVrcfQIJ1PJL0Bto46REC1ETLVkEz1XapkyV3VuYnWJpW2TWAd0nz
/dev/disk/by-id/dm-uuid-LVM-
bCUwiDLYkH0OVrcfQIJ1PJL0Bto46REC5lT9ZpQAj5GBy7aLPc6uFcoUXewEk9qc
/dev/disk/by-id/dm-uuid-LVM-
bCUwiDLYkH0OVrcfQIJ1PJL0Bto46RECNLILKPxBMPc0GVO3Ji2CwX4Hyi1rq5dp
/dev/disk/by-id/dm-uuid-LVM-
bCUwiDLYkH0OVrcfQIJ1PJL0Bto46RECQ0d4p6FBF6YLJe2icAX1LfuRf9MZNiPo
/dev/disk/by-id/dm-uuid-LVM-
bCUwiDLYkH0OVrcfQIJ1PJL0Bto46RECdg4bxGoOxoGmBVgeiK1NAzUKbtBTattn
/dev/disk/by-id/lvm-pv-uuid-A83I6e-23Ps-ayZv-ivXO-GSgl-rLPv-AQzgOT
E: No known /dev/disk/by-id

STEP-4 ==========================================================
ROOTFS1 -- Syncing ISO contents

I: -------------------- Running pre upgrade scripts (/tmp/APOLLO_PLATFORM-
nsg-upgrade/nsg-upgrade/root/usr/share/ostree-upgrade/run-scripts/pre.d)
I: -------------------- Finished pre upgrade scripts

I: -------------------- transfer begin --------------
I: Fetching '/tmp/APOLLO_PLATFORM-nsg-upgrade/iso-mounts/APOLLO_PLATFORM-
3.21.3.0-7+auto15/rootfs/usr/share/product-defs/product.conf'
I: Project='APOLLO_PLATFORM'
```

Unpack the debirf ISO

Unpack the contents of the debirf live ISO

- Mount the image

```
sudo mount -o loop debirf-live_bullseye_amd64.iso /mnt/iso
```

- Create a destination for the compressed files

```
mkdir ~/debirf
```

```
Copy the initrd file to the new directory
```

```
cp /mnt/iso/debirf-live_bullseye_6.0.0-0.deb11.6-amd64.cgz ~/debirf
```

```
cd ~/debirf
```

Change the extension of the file to be recognized by gzip

```
mv debirf-live_bullseye_6.0.0-0.deb11.6-amd64.cgz debirf-
live_bullseye_6.0.0-0.deb11.6-amd64.gz
```

- Uncompress the file

```
gunzip debirf-live_bullseye_6.0.0-0.deb11.6-amd64.gz
```

- Unpack the cpio file

```
cpio -idv < debirf-live_bullseye_6.0.0-0.deb11.6-amd64
```

- Change the extension of rootfs to be recognized by unxz

```
mv rootfs.cxz rootfs.xz
```

Uncompress the rootfs file

```
unxz rootfs.xz
```

```
Create a new directory to store the rootfs
```

```
mkdir rootdir
```

```
cd rootdir
```

Unpack the cpio file

```
cpio -idv < ../rootfs
```