ComposeFS and Containers

Fosdem 2024

Alexander Larsson - alexl@redhat.com
What is even ComposeFS?

"an opportunistically sharing verified image filesystem"
ComposeFS by example - creating an image

```bash
# tree rootfs/
rootfs/
    ├── foo.txt
    └── subdir
        └── bar.txt
# mkcomposefs --digest-store=objects rootfs example.cfs
```
ComposeFS by example - source files

Resultant files:

- example.cfs
- objects
  - 21
    - de27314f37
  - de
    - 2df1b63909

Content:

- objects/de/2df1b63909: foo
- objects/21/de27314f37: bar
ComposeFS by example - mounting

# mount -t composefs -o basedir=objects example.cfs mnt
# tree mnt/
mnt/
    ├── foo.txt
    │    └── subdir
    │         └── bar.txt
# cat mnt/foo.txt
foo
# cat mnt/subdir/bar.txt
bar
ComposeFS by example - role of base dir

# cat objects/de/2df1b63909
foo
# echo not-foo > objects/de/2df1b63909
# mount -t composefs -o basedir=objects example.cfs mnt
# cat mnt/foo.txt
not-foo

**Important**: base directory is shared between different images

- Objects are content-addressed
- “opportunistically sharing”
  - Shared disk space
  - Shared in page cache
  - Use less network bandwidth
Intermission: What is fs-verity

- Enabling fs-verity on a file
  
  ```
  # fsverity enable a-file.txt
  # cat a-file.txt
  content
  ```

- Makes file immutable (read-only)
  
  ```
  # echo foo >> a-file.txt
  a-file.txt: Operation not permitted
  ```

- Also validated by checksum (merkle tree hash)
  
  ```
  # fsverity measure a-file.txt
  sha256:8de6a0ac1443 a-file.txt

  # cat a-file.txt
  fs-verity (vda1, inode 12): FILE CORRUPTED! pos=0, level=-1,
  want_hash=sha256:43bb47cee21a, real_hash=sha256:d65b3c29f641
cat: a-file.txt: Input/output error
  ```
Weakness of fs-verity

- File contents can’t be changed
- But lots of things still can
  - File metadata
    - Permissions
    - Ownership
    - Setuid
  - Directory structure
    - New files
    - Renames
    - Replace files
- Need to validate an entire directory structure
Back to ComposeFS

# fsverity measure objects/21/de27314f37
sha256:21de27314f37 objects/21/de27314f37
ComposeFS using fs-verity

- We mount with the verity option

  ```
  # mount -t composefs -o basedir=objects,verity=on example.cfs mnt
  # ls -l mnt/
  -rw-r--r--. 1 root root  4 Aug 30 07:58 foo.txt
  drwxr-xr-x. 2 root root 46 Aug 30 07:59 subdir
  ```

- Image contains the expected digest of base files:

  ```
  # cat mnt/foo.txt
  overlayfs: lower file '2df1b639099a' has the wrong fs-verity digest
  cat: mnt/foo.txt: Input/output error
  ```
ComposeFS using fs-verity

- But what about the image file itself?

  ```bash
  # fsverity enable example.cfs
  # fsverity measure example.cfs
  sha256:0e693e188c example.cfs
  # mount -t composefs -o basedir=objects,verity=on,digest=0e693e188c example.cfs mnt
  ```

- Mount fails if wrong or no digest:

  Failed to mount composefs example.cfs: Image has wrong fs-verity
  Failed to mount composefs example.cfs: Image has no fs-verity

- **Root of trust**: toplevel composefs image digest
ComposeFS implementation details

- Initially a new kernel filesystem
- Now based on existing technologies
  - Overlayfs
  - Erofs
- Overlayfs layers
  - objects layer - basedir
  - Metadata layer - erofs loopback
    - Directory structure
    - File metadata
    - Redirects to lower layer for file content
- New overlayfs features
  - Data-only lower directories (merged in 6.5)
  - Fs-verity validation of redirects (merged in 6.6)
  - Nested overlay mounts (merged in 6.7)
- Userspace 1.0 version released
  - Supported stable image format
ComposeFS integration with Ostree

- Ostree is an image-based operating system model
- On disk format very similar to composefs
- Used by Fedora Silverblue
- Latest Ostree version:
  - Supports generating and mounting composefs image
  - Supports signing of composefs digest
  - Verifies composefs signature from Initrd
- In combination with SecureBoot this can give a fully tamper-proof OS
What about container images?

- Composefs also targets OCI images
- Work by Giuseppe Scrivano
  - Based on work on zstd:chunked
  - containers/storage - has basic composefs support
  - Podman - Needs vendor update to containers/storage
- Will allow
  - Higher container density due to file sharing
    - Memory (page cache)
    - Disk
  - Validation of images
    - Protects against accidental modifications
    - In future will allow signatures
How to use

- Enable in /etc/containers/storage.conf:

  ```
  [storage.options]
  pull_options = {enable_partial_images = "true",
                  use_hard_links = "true",
                  convert_images = "true"}
  
  [storage.options.overlay]
  use_composefs = "true"
  ```

- Use zstd:chunked images for better performance
Traditional overlayfs storage model

```
/var/lib/containers/storage
  └── overlay
      └── e7c4d59b867750b9a4e2f20eee83d044e2292177d325426e10cc56d9d3dae666
      └── diff
          ├── bin
          │   └── zcip
          │           ├── addgroup
          │           │       ... binaires...
          │           └── zcip
          ├── dev
          └── etc
              ├── groups
              │       └── shadow
              └── passwd
          └── tmp
          └── home
          └── root
          └── var
              └── usr
                  └── bin
                      └── env -> ../../bin/env
```
Composefs container storage model

/var/lib/containers/storage
  └─ overlay
      └─ e7c4d59b867750b9a4e2f20eee83d044e2292177d325426e10cc56d9d3dae666
        └─ composefs-data
            └─ composefs.blob
                └─ diff
                    ├── 46
                    │   └── 6afb852e38d454b87ab903abd189ea4541bd79bdf15449ccce7460af94d711
                    │   └── 78
                    │       └── 54ab1e3e8117a62cf74f4f0e161c24e425ef6a697ae7d16847729678c8e1df
                    │   └── 7f
                    │       └── d98179c77cddf6db0eb5041df3049d834cae41598dccda903ee3c30c43d833
                    │   └── 8b
                    │       └── 85846791ab2c8a5463c83a5be3c043e2570d7448434d41398969ed47e3e6f2
                    ├── b2
                    │   └── d68324f72d53c42b64121df172ff36c568a391db7236132a749cecdbe45cd
                    └── f5
                        └── 5824ead3d8f552bc22020211a8b181af4506e4fbba20389114e46c1cefd9c
Resource use example

- Install 20 almost identical images
  - All images are 1 layer
  - Only one file differs in all images
- Run “sleep” in each of them
- Simulates 20 containers that use same glibc package

```bash
for i in `seq 20`; do
  bin/podman run -d docker.io/alexl/shared:$i \ 
    sleep 100000
done
```
Disk use - legacy

```
# du -csh /var/lib/containers/storage/overlay/*
181M /var/lib/containers/storage/overlay/1b16136ca74aeb83c6f6e43cd1cb13b8f029d4db0fa183a8910d5ecb6eddb2a51
181M /var/lib/containers/storage/overlay/202ea5130da11b511b9c529a487e483133d9669b7145b5972d98073de86b2dd7
181M /var/lib/containers/storage/overlay/29f94f291aeb0fd5fd34ad13d40565545c58cd2d554736b4c44232b0b30b450a52
181M /var/lib/containers/storage/overlay/2f0ec1fe5185810b539cafc8522d6280947a0ad66bb7ef19931006445a83e8
181M /var/lib/containers/storage/overlay/40f656900f7b9a98f757a581e1d9638a5327749f51a5e101f14010b73a16b7cf3
181M /var/lib/containers/storage/overlay/4864a23ce1c7651c91387ae7f908f5622ff9a652c69f494a129870afc10051
181M /var/lib/containers/storage/overlay/4f528697a115ccf4cb4c1b25a2a6912f1b6ec31f3f8bce33b9f5d3d6401f
181M /var/lib/containers/storage/overlay/5c78519af846cfc55e7b74654a28a2707e4d1e5fd31d021b5364545d79b7134
181M /var/lib/containers/storage/overlay/6acadd22c853bd605b65d19eed9c199112af0d07b9abc9ae4c1662823c4ec83
181M /var/lib/containers/storage/overlay/8aa5f13fed227606494a871644aa80dc56e321163b25cad1ba83f7c1c5991abf
181M /var/lib/containers/storage/overlay/a181ee05798c946315e4c5b6064c370a75f4d47fa81a063e523409ea798ee15a
181M /var/lib/containers/storage/overlay/a86db69af5968e67c0db161e3ed808ee8379abec9022a7d417887491c5d
181M /var/lib/containers/storage/overlay/abc5a821329e145ec16d2f73d943f18a92be03ad03cf958b3567fd5b4286d6c7
181M /var/lib/containers/storage/overlay/b24faebee037b581b3e5484e8d749074f14f3522991e336e3d2642a1146682abe
181M /var/lib/containers/storage/overlay/b992a9f8eac6b7b730bdeee0b9fbed8580ba3a2cca0129361f1fb30279d5268
181M /var/lib/containers/storage/overlay/d63c4796ee5c070980c9729bc5f183d11781ca34d6c05a4328d3e4236b524d4
181M /var/lib/containers/storage/overlay/d787bcb864b28639c4649b4abbb531f1aa797c2e8f1d184a2d8949a6e6a5e57ad
181M /var/lib/containers/storage/overlay/d831373ac4fd6b60ca7b08d8732140077e5dca8e8b2562d4896f3e3cb5309b
181M /var/lib/containers/storage/overlay/ddb15cb5b3c3f148e74b99eb08961c6fd301b448a99d1c7aa7fbbf79587d9137b3
181M /var/lib/containers/storage/overlay/e18034bd9afbe1b6813e2bd5234f7ad9c900ebec56712530f8ecb0ab30e55f
3,6G  total
```
Disk use - composefs

```
# du -csh /var/lib/containers/storage/overlay/*
179M /var/lib/containers/storage/overlay/1b16136ca74ae83c6f6e43cd1cb13b8f0294db0fa183a8910d5ecb6ed6b2a51
179M /var/lib/containers/storage/overlay/202ea5130da11b511abc54aaf37e483133ad969b7145b5972d98073de886b2dd7
2,4M /var/lib/containers/storage/overlay/29ffdf291a8ebd5d5f3a4d1340f5f54c58cd2d554736d4c42cb03b30b450a52
2,4M /var/lib/containers/storage/overlay/2f0ec1fe5f15880b539caef8522c6289047a0ad66bb7ef199310064454a83e8
2,4M /var/lib/containers/storage/overlay/40f656900f7b9a9f877a581e1d9638a53277495f1a5e101f41007b3a16b7cf3
2,4M /var/lib/containers/storage/overlay/46b4a23ce1c7651c9138eae7fe908df622ff9a65269f494a129870aucf10051
2,4M /var/lib/containers/storage/overlay/4f528697a115cc4cb4c1b25a2ae69121b66ecf83b9f5df3d6401f
2,4M /var/lib/containers/storage/overlay/5c78519afdf84c6f55e7b74635a28a2707e4d1e5fd31d021b5364545d79b7134
2,4M /var/lib/containers/storage/overlay/6acadd2ec853bd605b65d19eed9c199112af020b9ab2c9ae4c1662823c443c83
2,4M /var/lib/containers/storage/overlay/8aaaf13fed22760625afad912d39add25cad7a83f7f35991abf
2,4M /var/lib/containers/storage/overlay/a181ee05798c946315e4c5b6064c370a75df4d447fa81a6036e523409ea798eee15a
2,4M /var/lib/containers/storage/overlay/a8d6b69a596c668e6c70db161e3ed809ee87979929c02522d7d4178874c91c5d
2,4M /var/lib/containers/storage/overlay/abc5a82329e145e5d2f73d943ff39a92be03ad03cf95f8bb3567fd5b4286d6c7
2,4M /var/lib/containers/storage/overlay/b24faeee037581b3e548e87e54907474f145322991e303e3e26242a146682abe
2,4M /var/lib/containers/storage/overlay/b992af9beefc6a874063d0e7b3773d2bee0b0fbedcd8580ba3a2c0a0299316f1fb30279d5268
2,4M /var/lib/containers/storage/overlay/d63a479ef6c167098c9729bc5fb183d11781ca34d6c054a4328d34e2436b52a4d4
2,4M /var/lib/containers/storage/overlay/d787bc8642b28639c4649b4a5bb31f1a79772e81d184a2d8949a6a65e577adb
2,4M /var/lib/containers/storage/overlay/d83a1373ac4f6dbb6a0ab708b8732140077e5dca8e8b2562d4389f63e3cb5309b
2,4M /var/lib/containers/storage/overlay/ddb15c5bc3f14f874b99eb08961cf6d301b448a99d1c7aa7f7f9587d9137b3
2,4M /var/lib/containers/storage/overlay/e1803f4bd9afbecca1b6813e2bd523fe7adc990beec56712530ffe0bca0b30e55f
224M total
```
## Memory use - legacy

```
# smem -P sleep
PID User  Command   Swap  USS    PSS    RSS
  1361786 root sleep 100000 0 1376 1376 1380
  1360881 root sleep 100000 0 1444 1444 1448
  1361178 root sleep 100000 0 1496 1496 1500
  1360979 root sleep 100000 0 1548 1548 1552
  1361738 root sleep 100000 0 1580 1580 1584
  1361237 root sleep 100000 0 1584 1584 1588
  1361633 root sleep 100000 0 1588 1588 1592
  1361287 root sleep 100000 0 1596 1596 1600
  1361681 root sleep 100000 0 1596 1596 1600
  1360831 root sleep 100000 0 1600 1600 1604
  1360930 root sleep 100000 0 1600 1600 1604
  1361027 root sleep 100000 0 1600 1600 1604
  1361081 root sleep 100000 0 1604 1604 1608
  1361130 root sleep 100000 0 1604 1604 1608
  1361385 root sleep 100000 0 1604 1604 1608
  1361485 root sleep 100000 0 1604 1604 1608
  1361534 root sleep 100000 0 1604 1604 1608
  1361436 root sleep 100000 0 1640 1640 1644
  1361335 root sleep 100000 0 1656 1656 1660
  1361584 root sleep 100000 0 1656 1656 1660
```

**USS** - Unique Set Size  
**PSS** - Proportional Set Size  
**RSS** - Resident set size
## Memory use - composefs

### # smem -P sleep

<table>
<thead>
<tr>
<th>PID</th>
<th>User</th>
<th>Command</th>
<th>Swap</th>
<th>USS</th>
<th>PSS</th>
<th>RSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1364330</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>96</td>
<td>160</td>
<td>1380</td>
</tr>
<tr>
<td>1364734</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>100</td>
<td>164</td>
<td>1384</td>
</tr>
<tr>
<td>1364882</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>100</td>
<td>164</td>
<td>1384</td>
</tr>
<tr>
<td>1364630</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>96</td>
<td>166</td>
<td>1448</td>
</tr>
<tr>
<td>1364983</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>96</td>
<td>168</td>
<td>1504</td>
</tr>
<tr>
<td>1364379</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>96</td>
<td>170</td>
<td>1452</td>
</tr>
<tr>
<td>1364579</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>92</td>
<td>171</td>
<td>1548</td>
</tr>
<tr>
<td>1364478</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>100</td>
<td>174</td>
<td>1456</td>
</tr>
<tr>
<td>1364428</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>92</td>
<td>177</td>
<td>1600</td>
</tr>
<tr>
<td>1364684</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>100</td>
<td>179</td>
<td>1556</td>
</tr>
<tr>
<td>1364121</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>96</td>
<td>180</td>
<td>1608</td>
</tr>
<tr>
<td>1364171</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>96</td>
<td>180</td>
<td>1608</td>
</tr>
<tr>
<td>1364529</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>96</td>
<td>180</td>
<td>1608</td>
</tr>
<tr>
<td>1364280</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>96</td>
<td>181</td>
<td>1600</td>
</tr>
<tr>
<td>1364932</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>96</td>
<td>181</td>
<td>1604</td>
</tr>
<tr>
<td>1364021</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>100</td>
<td>185</td>
<td>1604</td>
</tr>
<tr>
<td>1364220</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>100</td>
<td>185</td>
<td>1604</td>
</tr>
<tr>
<td>1364833</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>100</td>
<td>187</td>
<td>1592</td>
</tr>
<tr>
<td>1364782</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>108</td>
<td>197</td>
<td>1664</td>
</tr>
<tr>
<td>1364071</td>
<td>root</td>
<td>sleep 100000</td>
<td>0</td>
<td>120</td>
<td>205</td>
<td>1588</td>
</tr>
</tbody>
</table>

**Legend**

- **USS** - Unique Set Size
- **PSS** - Proportional Set Size
- **RSS** - Resident set size
Future work

- Complete current work and merge into podman
- Zstd: chunked by default
- Sign composefs images
  - Add digest composefs images to image metadata
  - Ensure image is signed
  - Validate signature and digest on run
  - Outstanding questions:
    - What key to sign with?
    - Where to store the public key?
    - What is the trust model for public key?
Questions

- https://github.com/containers/composefs
- https://github.com/containers/storage/