Container Storage Best Practices in 2017

Myth-busting and taking state of the present

Keith Resar
Red Hat Solution Architect
January 24th, 2017

Agenda

- Container Storage Myths
- Container Storage Primer
- Review 6 Storage Drivers
- Chooser a Storage Driver

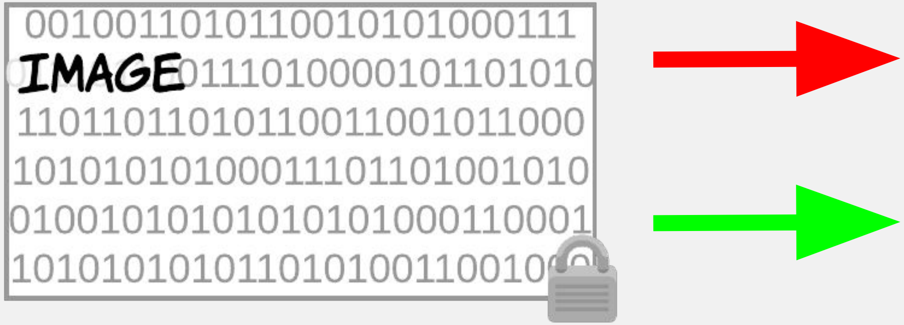
Container Storage History / Myths

http://en.wikifur.com/wiki/Sloth_(species)

Container Storage Level Set

Image : Container :: Class : Object

For humans, read this to say:
An image is to a container, as a class is to an object.

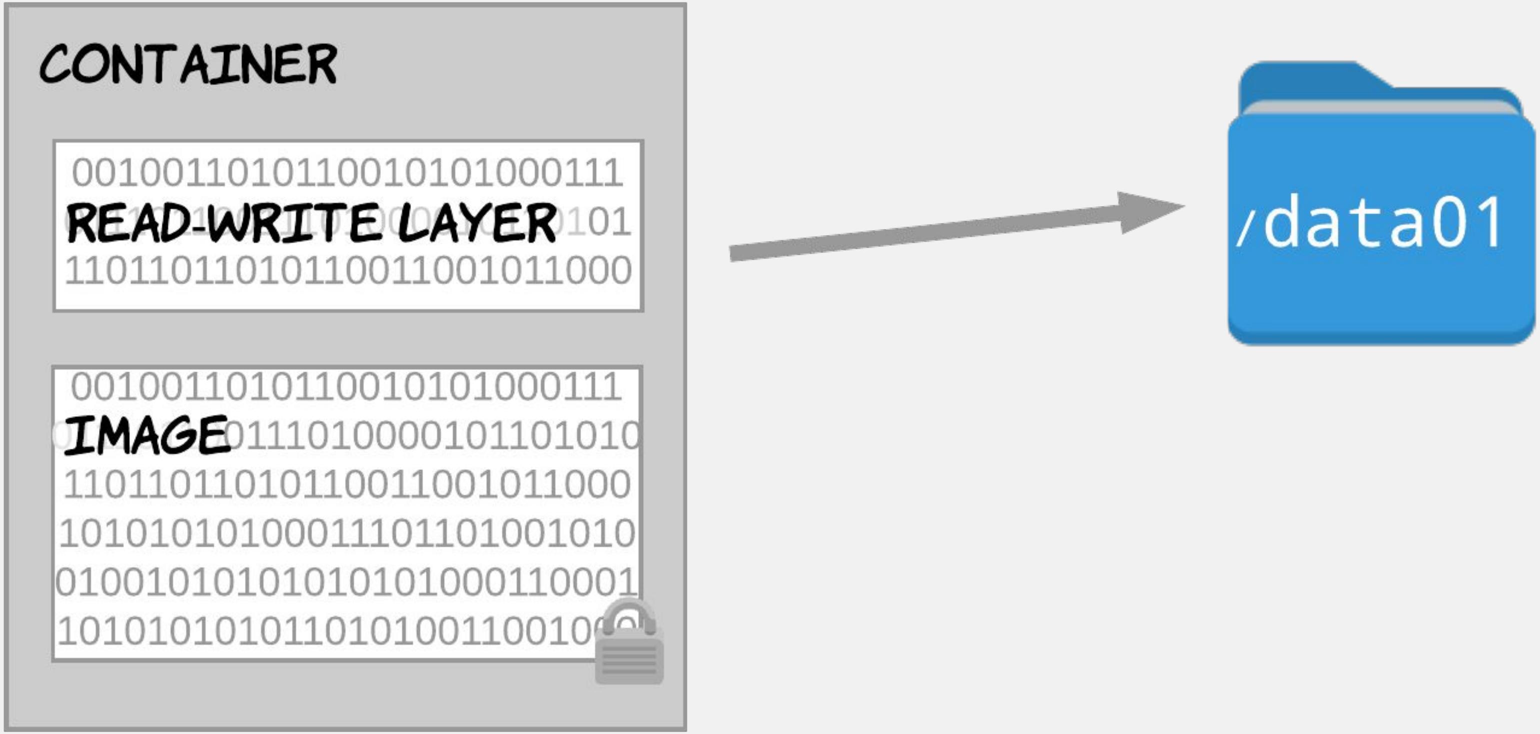


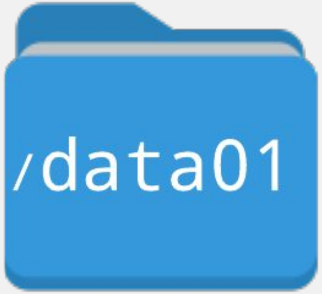
```
> ls -l /image; echo $?
0
```

```
> pgrep image; echo $?
1
```

Container Storage

Data Volume Storage





Data Volume Storage

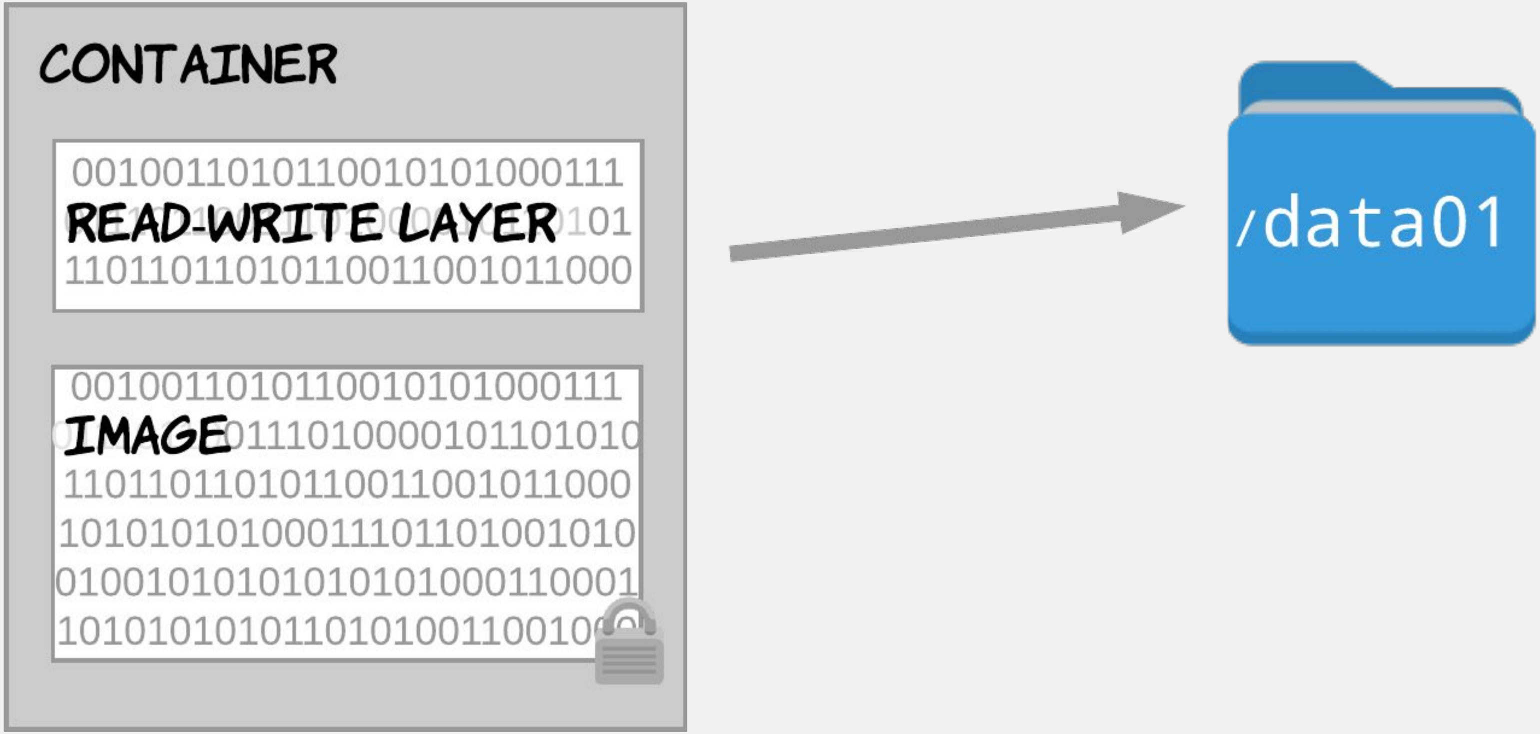
Can be extended to support many endpoints and protocols using installable docker plugins.

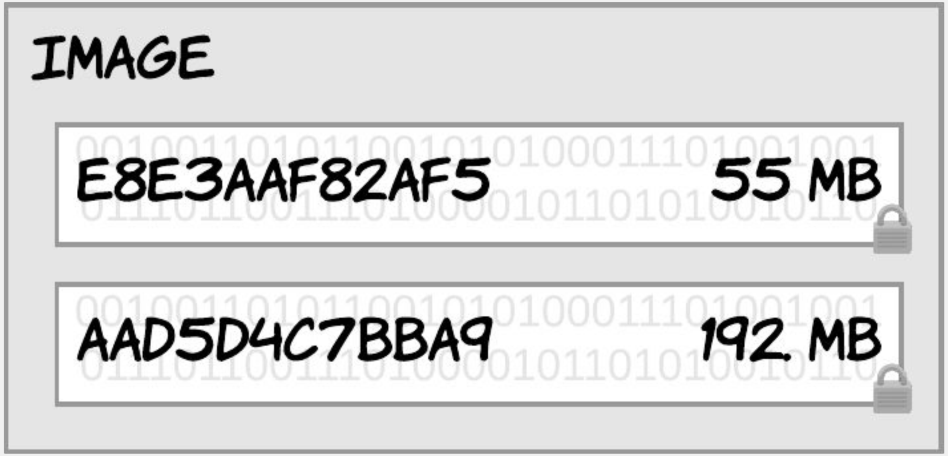
- Local
- LVM

- GlusterFS
- Ceph
- NFS
- iSCSI

Container Storage

Data Volume Storage





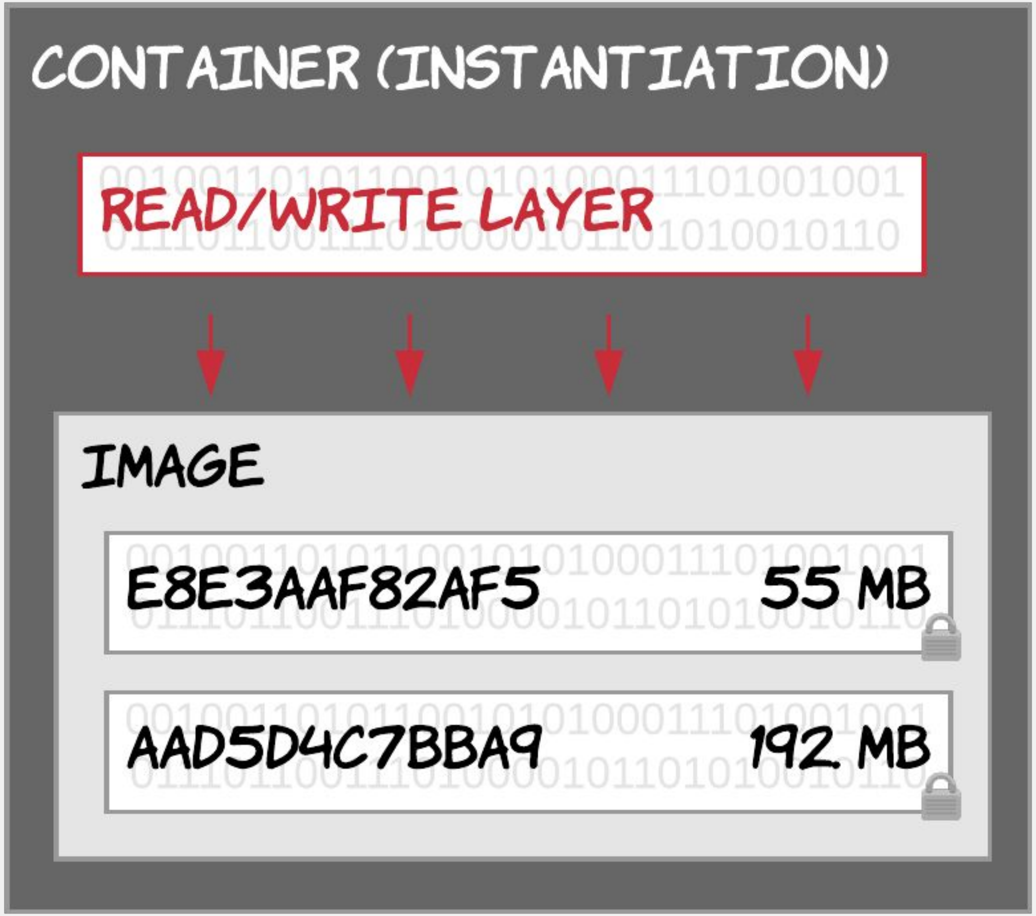
IMAGE

E8E3AAF82AF5 55 MB

AAD5D4C7BBA9 192 MB

httpd

rhel7:latest



CONTAINER (INSTANTIATION)

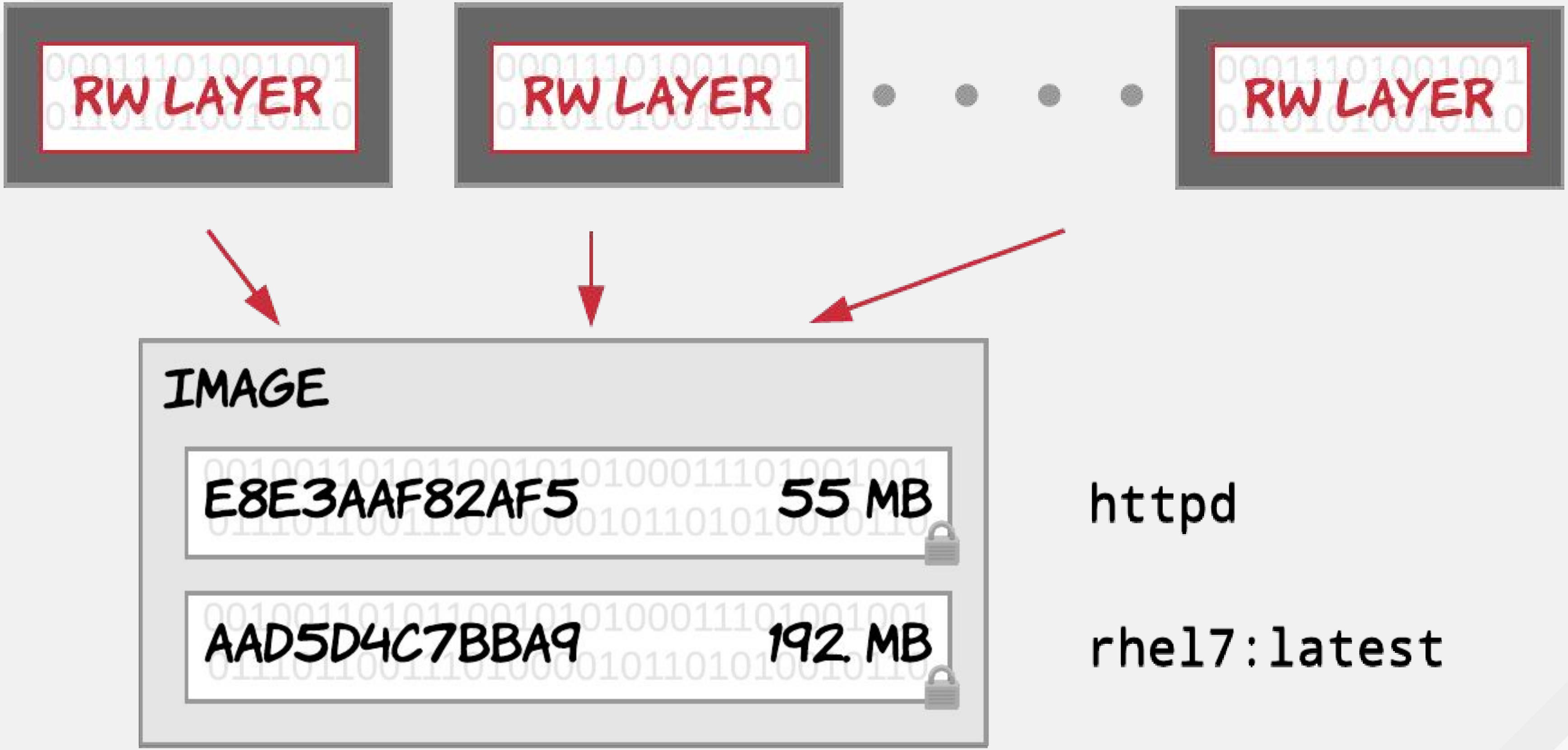
READ/WRITE LAYER

IMAGE

E8E3AAF82AF5 55 MB

AAD5D4C7BBA9 192 MB

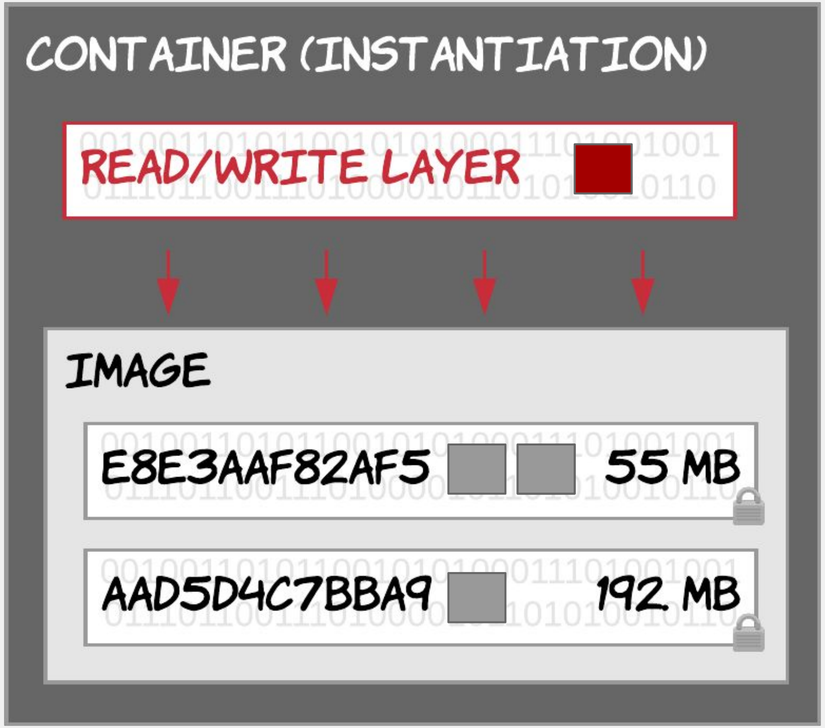
```
container layer

httpd

rhel7:latest
```



httpd

rhel7:latest

Copy-on-write Strategy



container layer

httpd

rhel7:latest

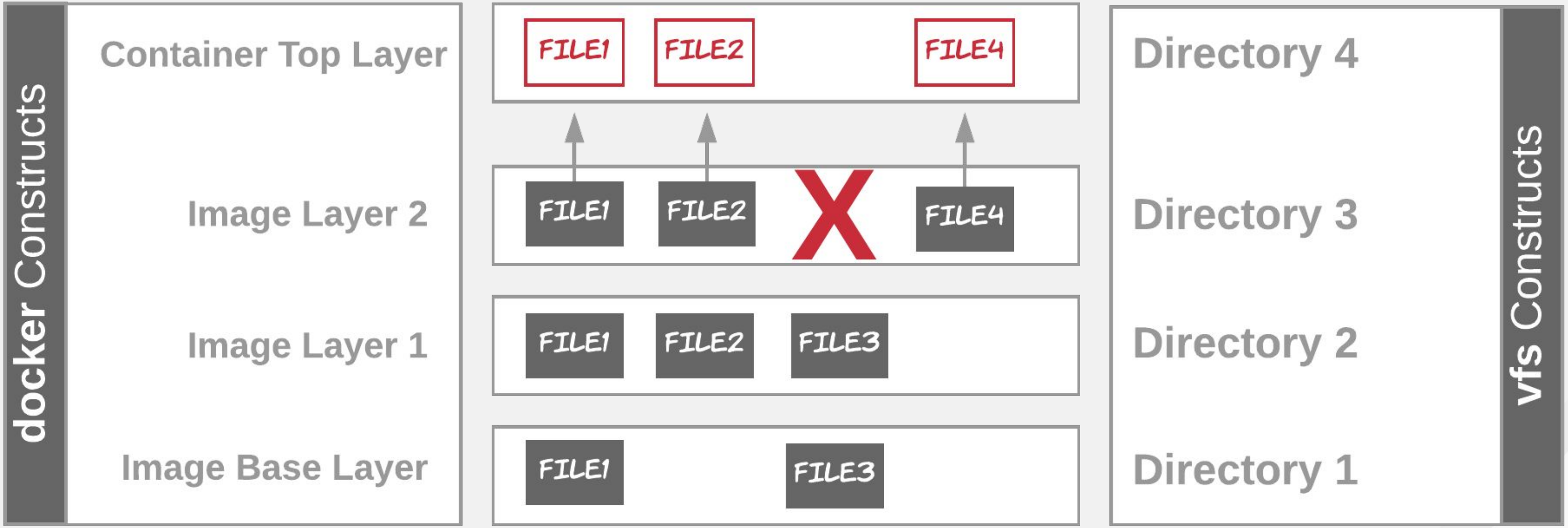
Container Storage Drivers

Available Storage Drivers

Technology	Driver	Introduction	File vs. Block
VFS	vfs	origin	* File
AUFS	aufs	origin	File
OverlayFS	overlay/overlay2	Aug 2014 (1.11) June 2016 (1.12)	File
Device Mapper	devicemapper	Sept 2013 (0.7)	Block
Btrfs	btrfs	Nov 2013	File

vfs Driver (1 of 6)

Naive implementation lacking union filesystem and copy-on-write



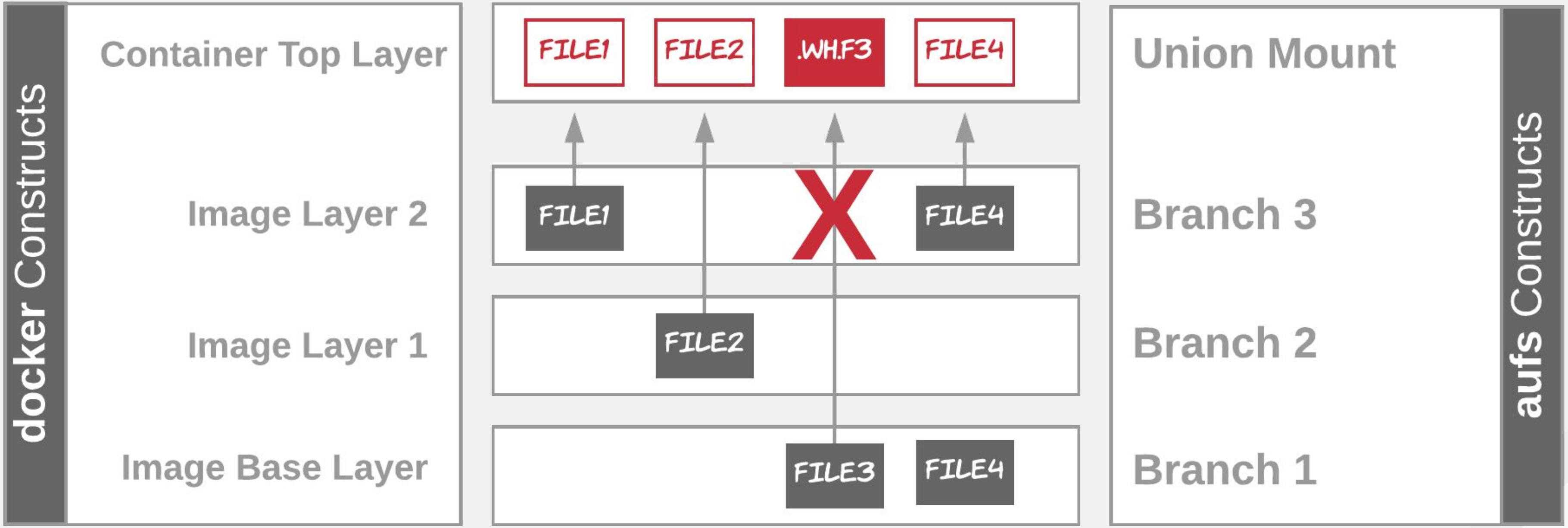
vfs Driver (1 of 6)

Naive implementation lacking union filesystem and copy-on-write

The Good	The Bad	Summary
Reference compatibility model Useful for docker-in-docker scenarios to avoid nesting storage drivers	No shared memory, union filesystem, or copy-on-write	Not for production use Important support role for storage driver development

AUFS Driver (2 of 6)

The original docker storage driver



https://docs.docker.com/engine/userguide/storagedriver/aufs-driver/

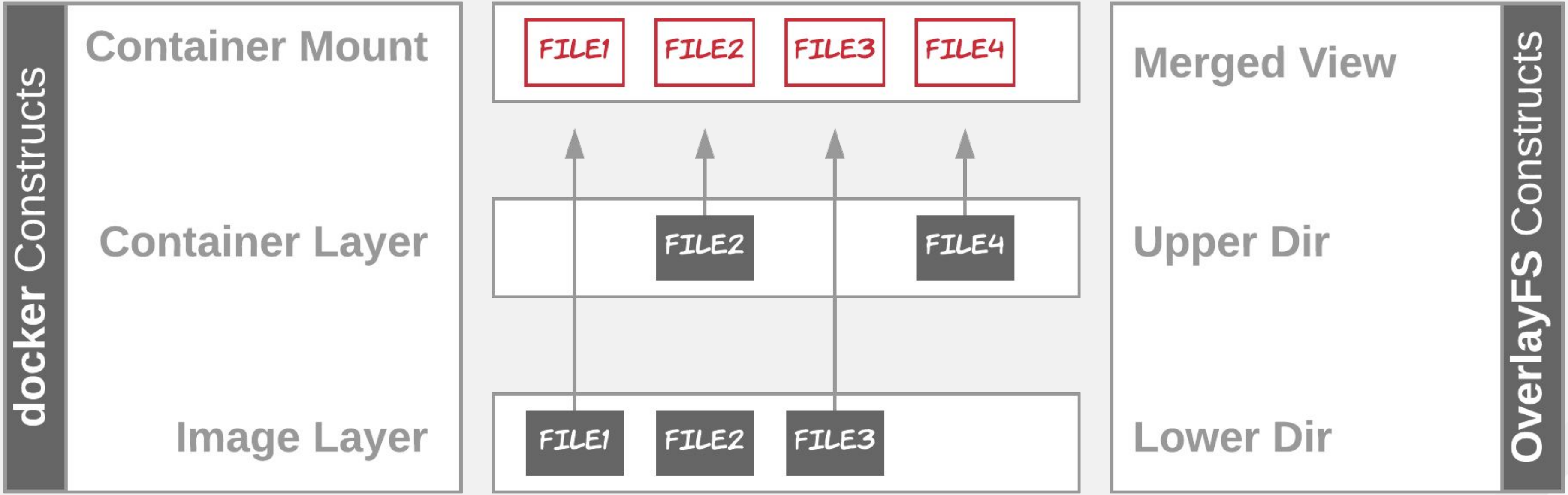
AUFS Driver (2 of 6)

The original docker storage driver

The Good	The Bad	Summary
Battle hardened driver Performant and stable for wide range of use cases Supports shared memory	Carried patch to mainline Linux kernel limits distro support File level implementation impacts copy-on-write	Default for non-RH, will meet majority of needs Expectation that it will be supplanted by an Overlay implementation

Overlay Driver (3 of 6)

Legacy union filesystem driver, superseded by overlay2



https://docs.docker.com/engine/userguide/storagedriver/overlayfs-driver/

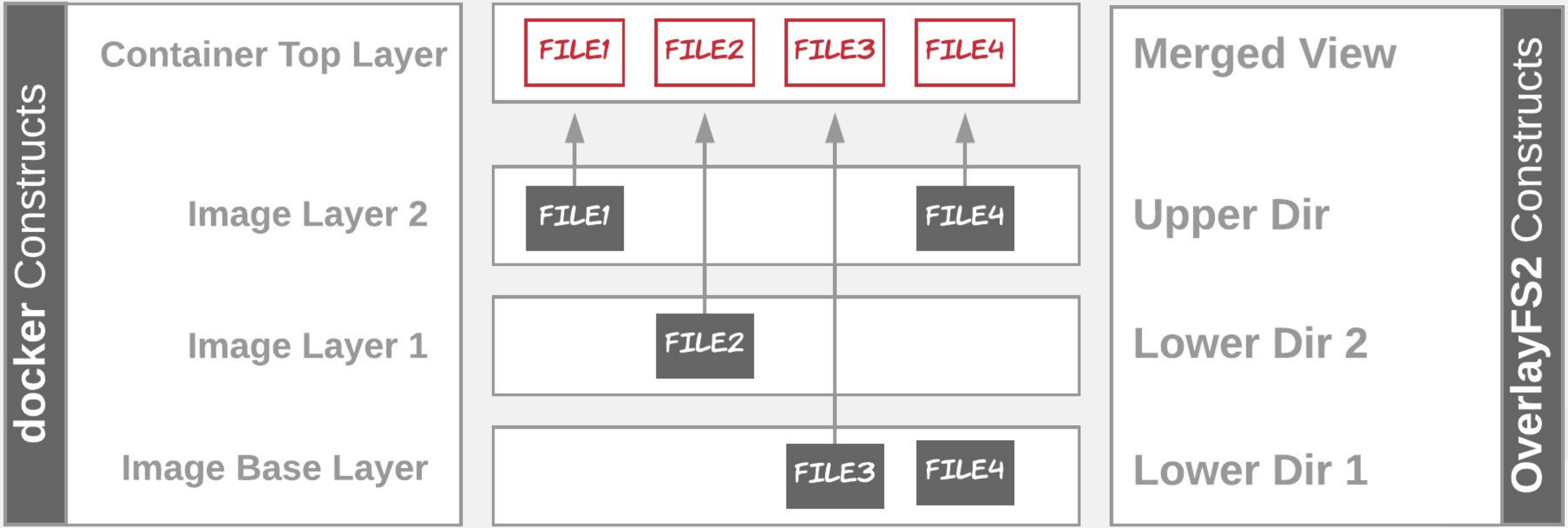
Overlay Driver (3 of 6)

Legacy union filesystem driver, superseded by overlay2

The Good	The Bad	Summary
Complete union filesystem merged into the mainline kernel Shared memory	Architecture drove explosive inode usage, often to the point of exhaustion Slow commit performance	Used for backward compatibility in pre-4.0 kernels Broad distro support beyond aufs

Overlay2 Driver (4 of 6)

Lessons learned from original overlay, and looking forward to continued maturity



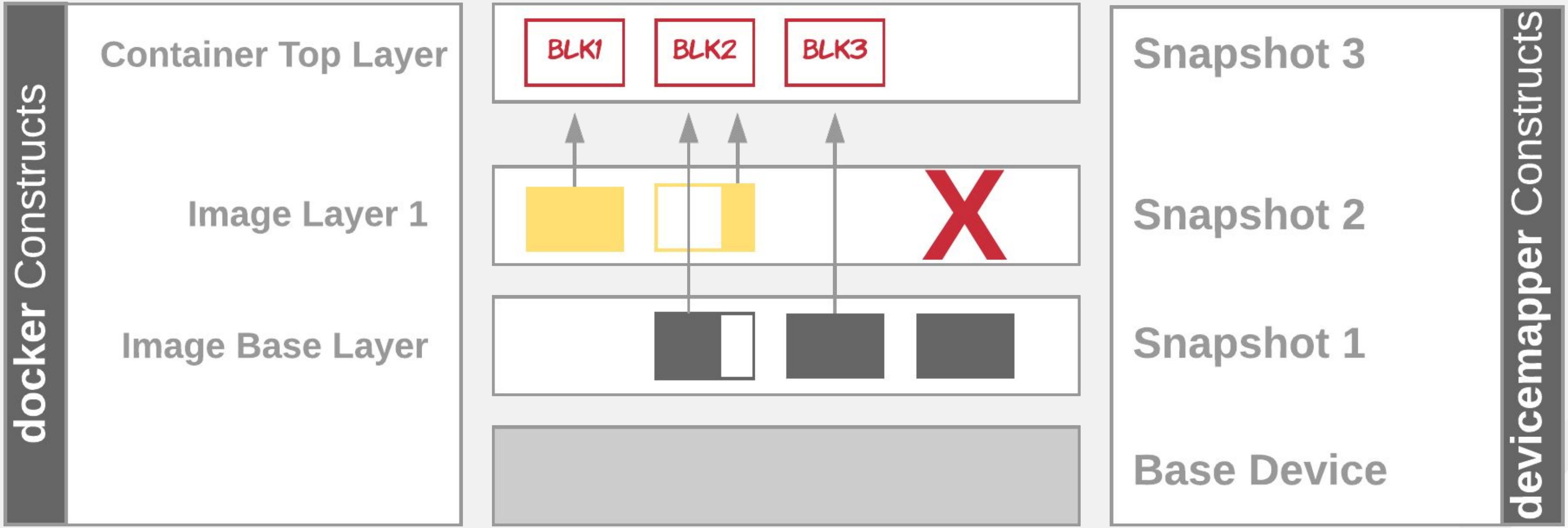
Overlay2 Driver (4 of 6)

Lessons learned from original overlay, and looking forward to continued maturity

The Good	The Bad	Summary
Retains all benefits of overlay (shared memory, broad distro support) Resolves inode exhaustion problems	Relatively young codebase (initial release with Docker 1.12 in June 2016) File-based so copy-on-write operations may be expensive	With maturity may be the best route forward for consistent defaults across many Linux distributions

Devicemapper Driver (5 of 6)

Lvm integrated block-based storage driver



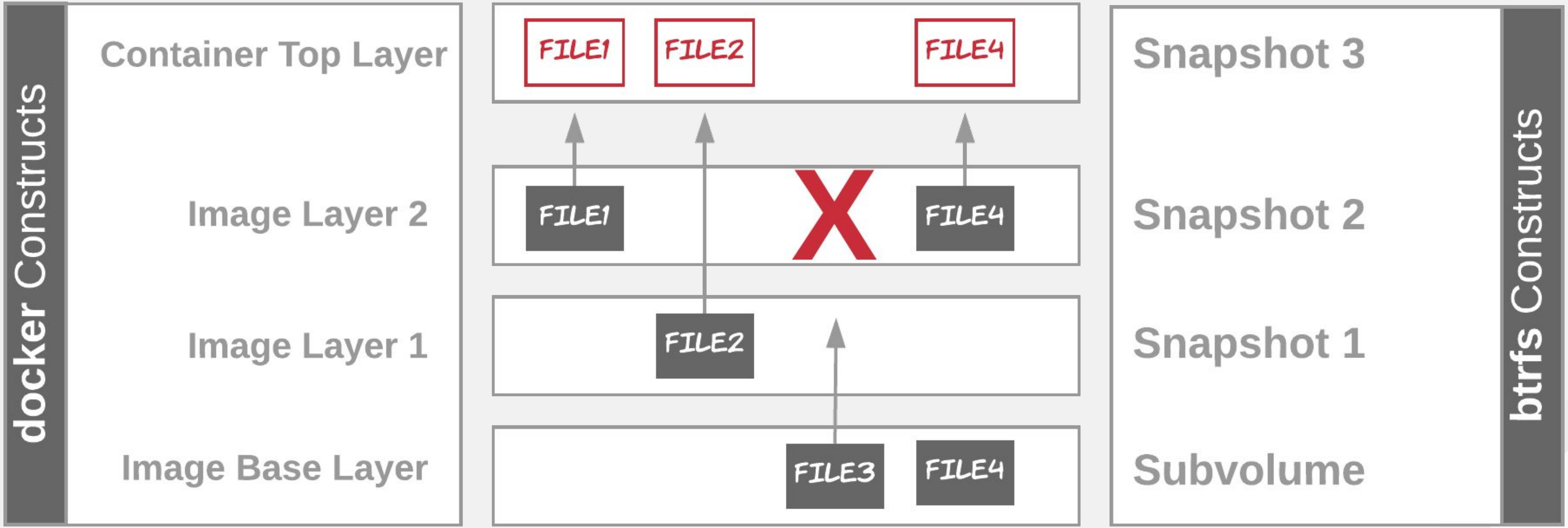
Devicemapper Driver (5 of 6)

Lvm integrated block-based storage driver, default on RHEL

The Good	The Bad	Summary
Block-based solution offers efficient copy-on-write Quota support Available direct and loop modes	Manual setup is intimidating No shared memory support	Red Hat go-to graphdriver with mature codebase

Btrfs Driver (6 of 6)

Another next generation filesystem, with a continued heavy development requirement



Btrfs Driver (6 of 6)

Another next generation filesystem, with a continued heavy development requirement

The Good	The Bad	Summary
Now offers SELinux support and quota	No page-cache sharing between containers Small writes can lead to out-of-space conditions Requires btrfs specific tools rather than Linux native	Btrfs hasn’t been a mainstream choice for Linux distros, driving less attention and less testing

Choosing a Storage Driver

Benchmark Approach

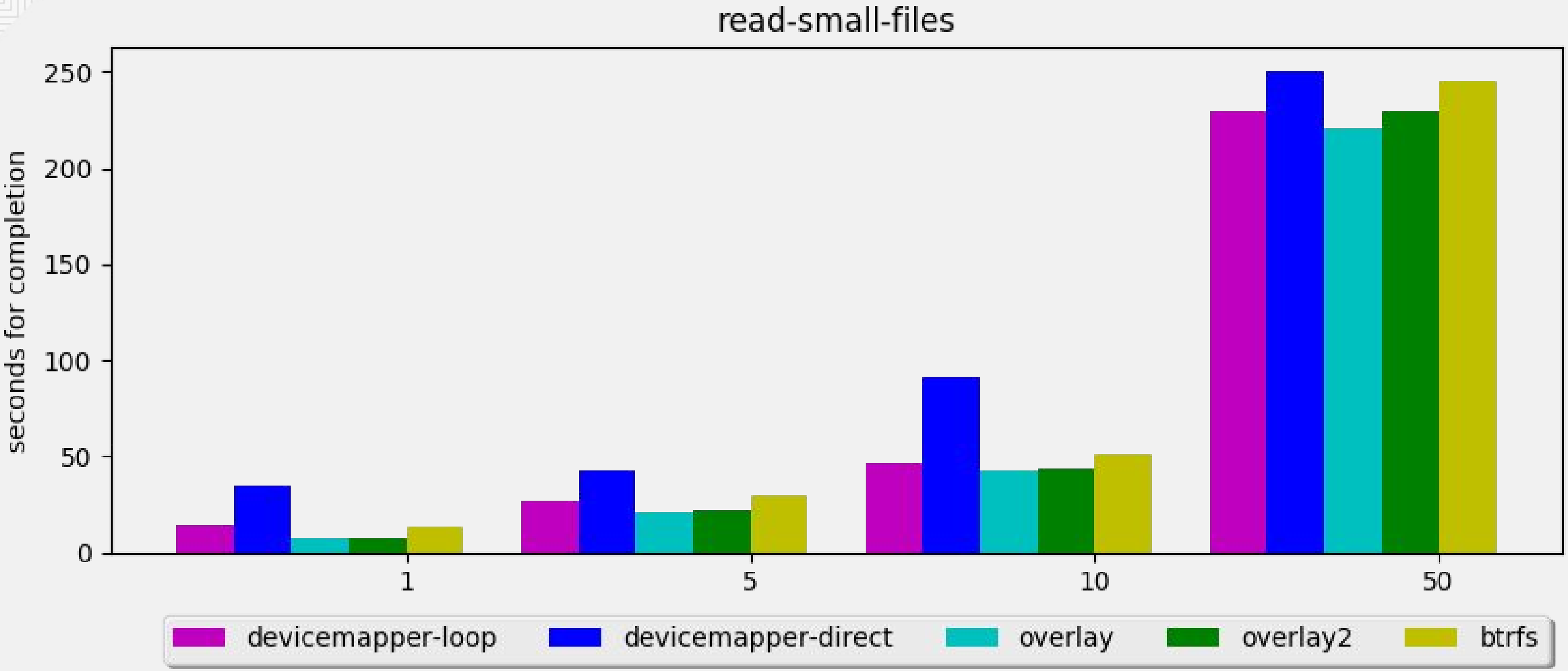
Benchmarking is treacherous and confusing, and often done poorly - which means that you need to take any benchmark results with a large grain of salt.

If you've spent less than a week studying a benchmark result, it's probably wrong.

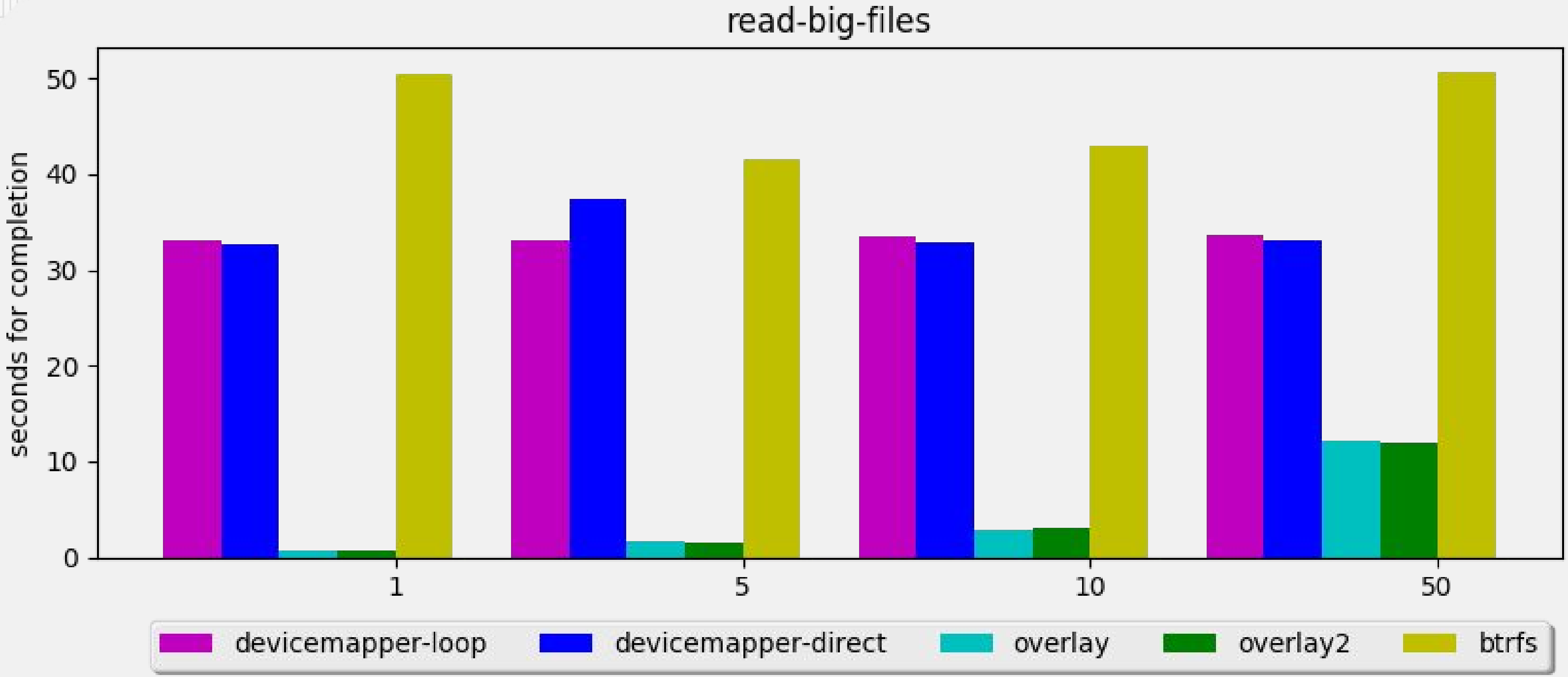
(Running a benchmark is the easy part. Understanding a benchmark can take much longer.)

Benchmark 1: Reading Files

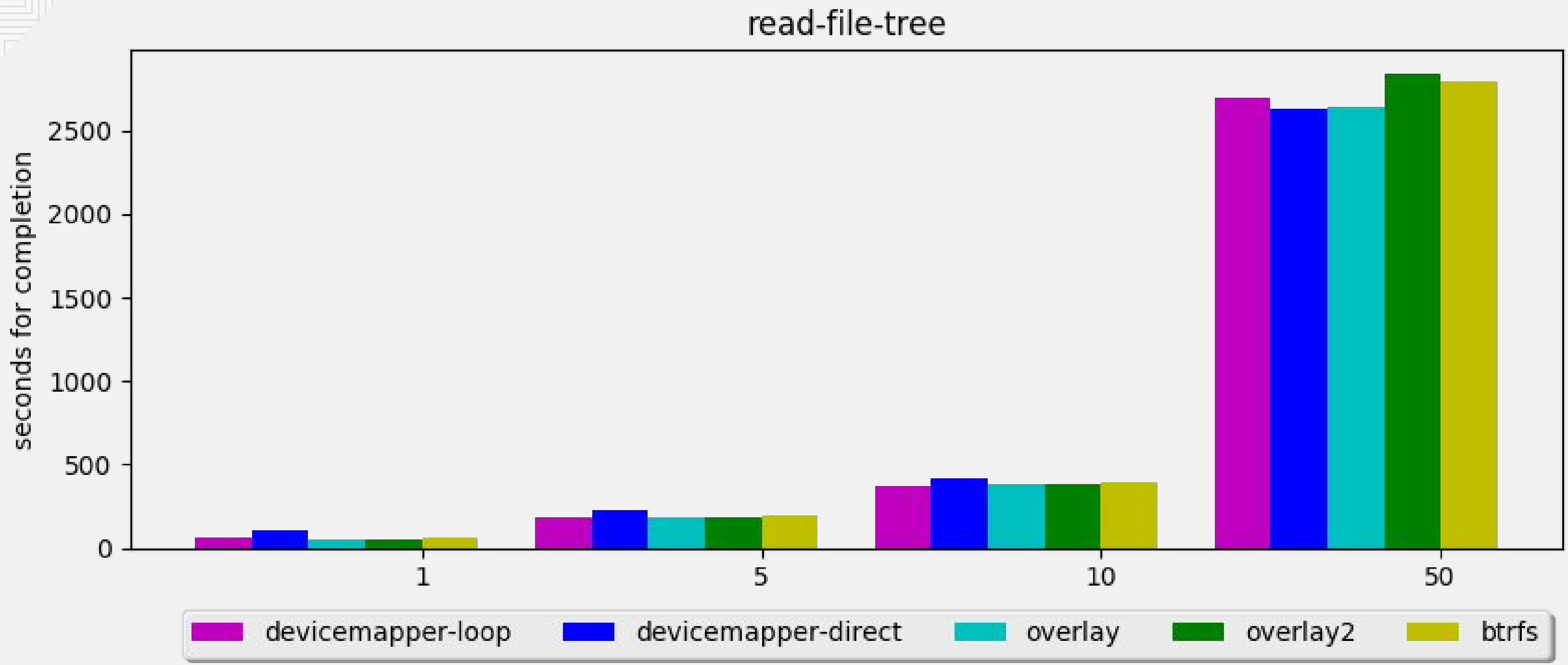
- Reading Small Files
- Reading Large Files
- Reading File Tree



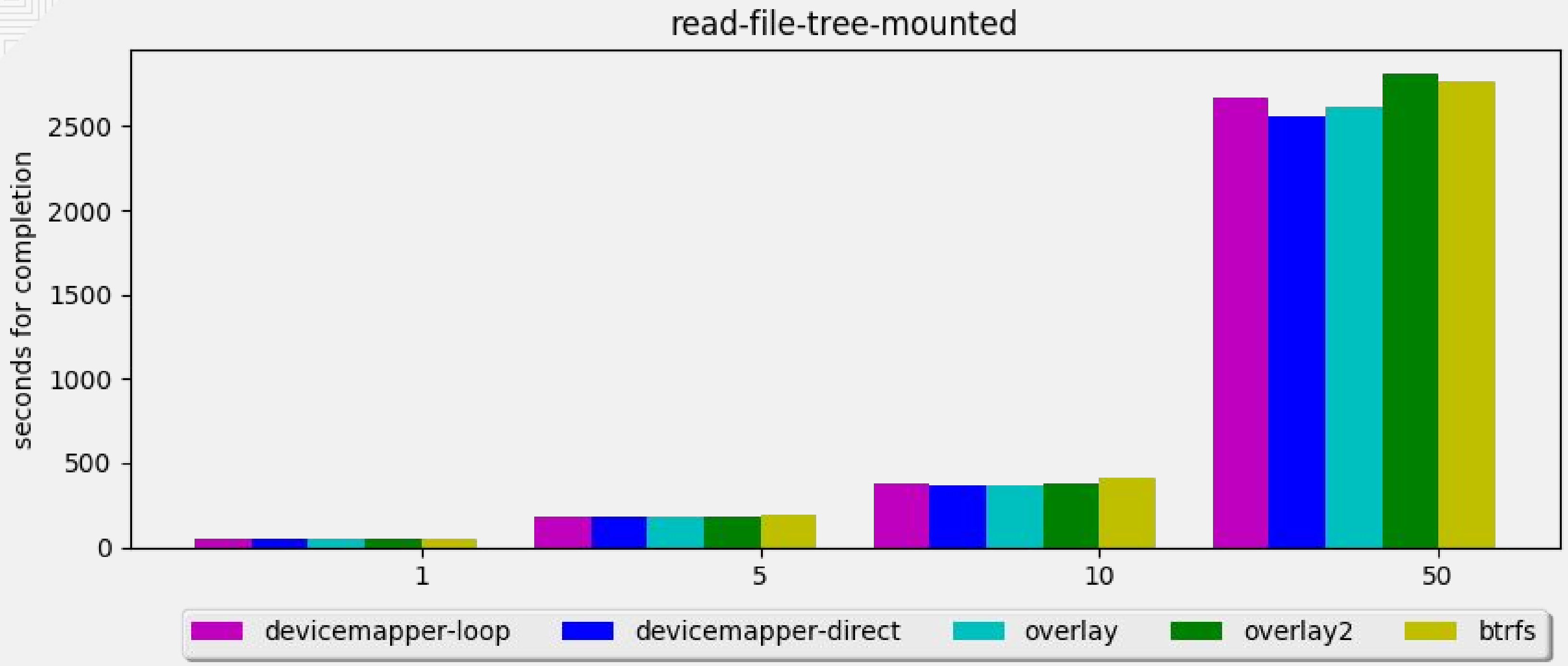
Naive benchmarking, for discussion purposes only. Don’t trust this!



Naive benchmarking, for discussion purposes only. Don’t trust this!



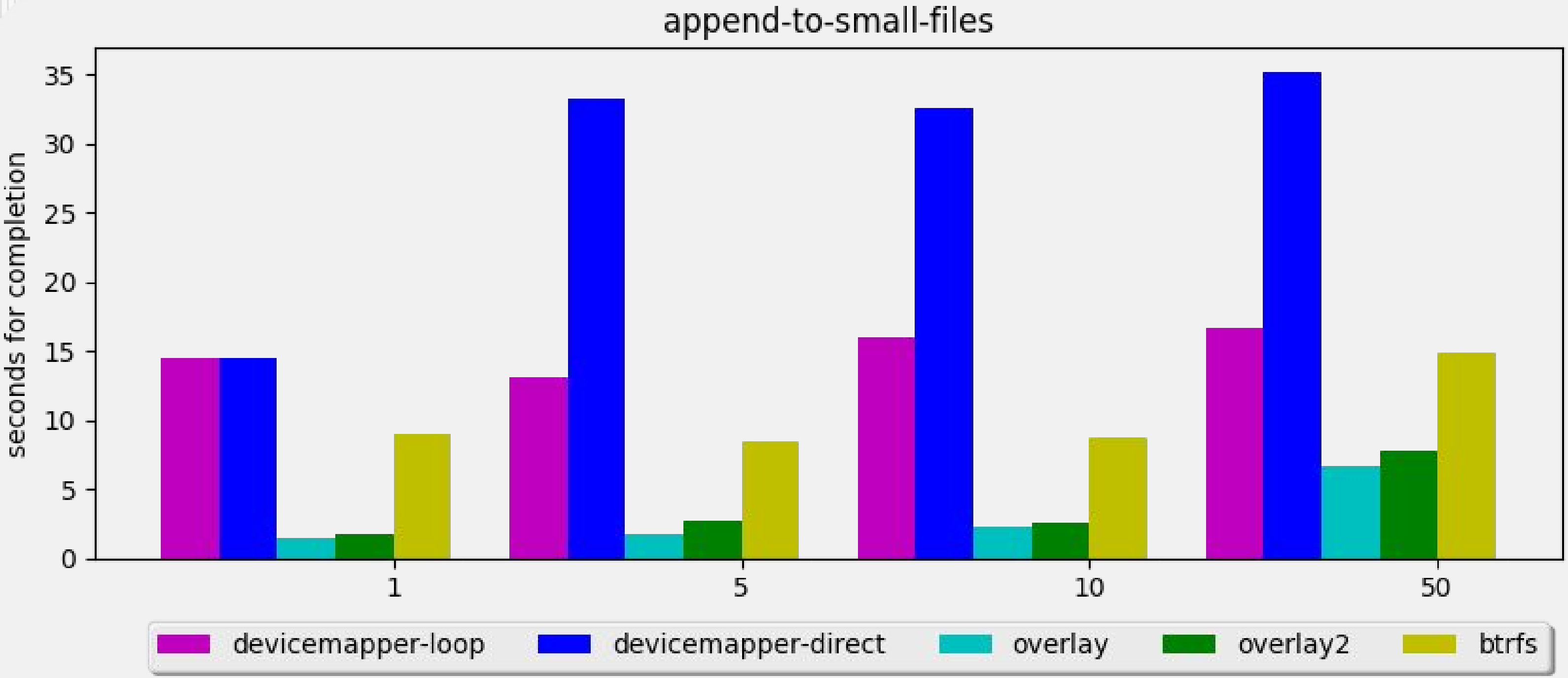
Naive benchmarking, for discussion purposes only. Don’t trust this!



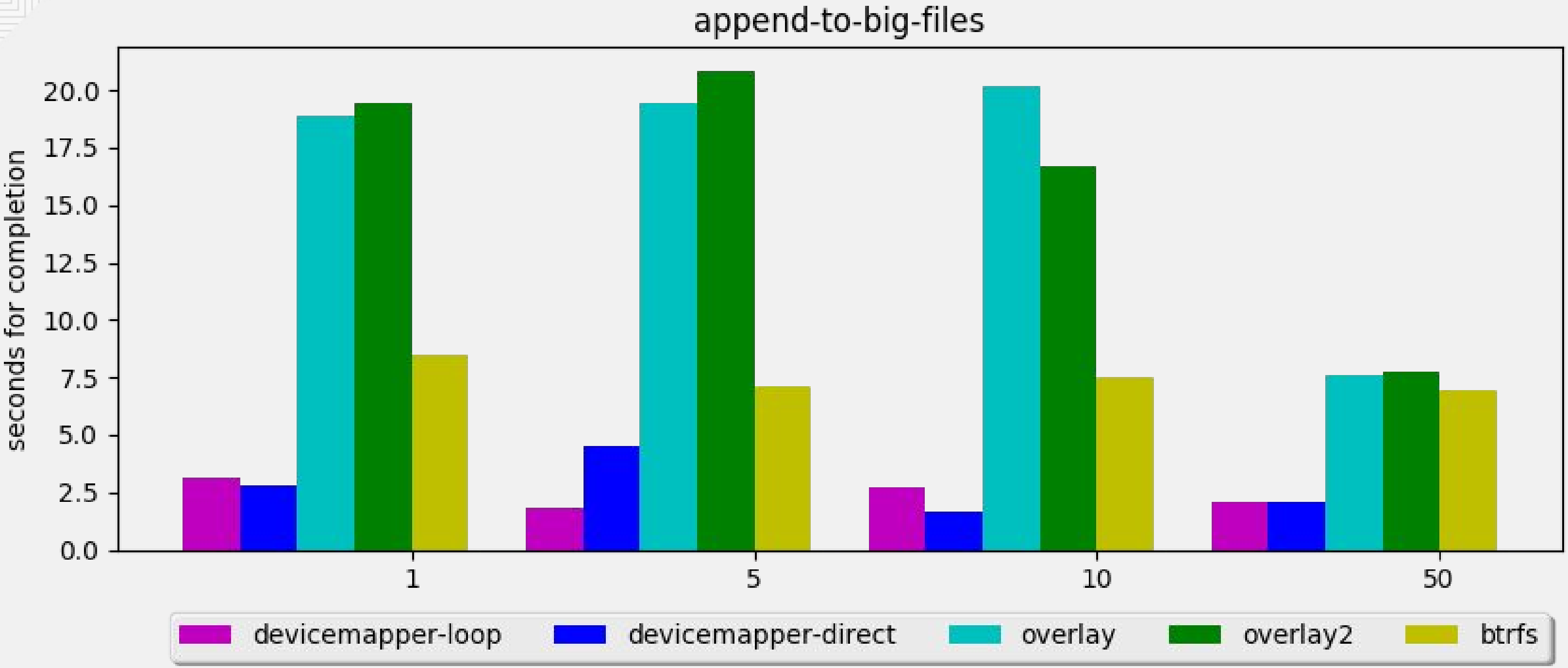
Naive benchmarking, for discussion purposes only. Don’t trust this!

Benchmark 2: Appending to Files

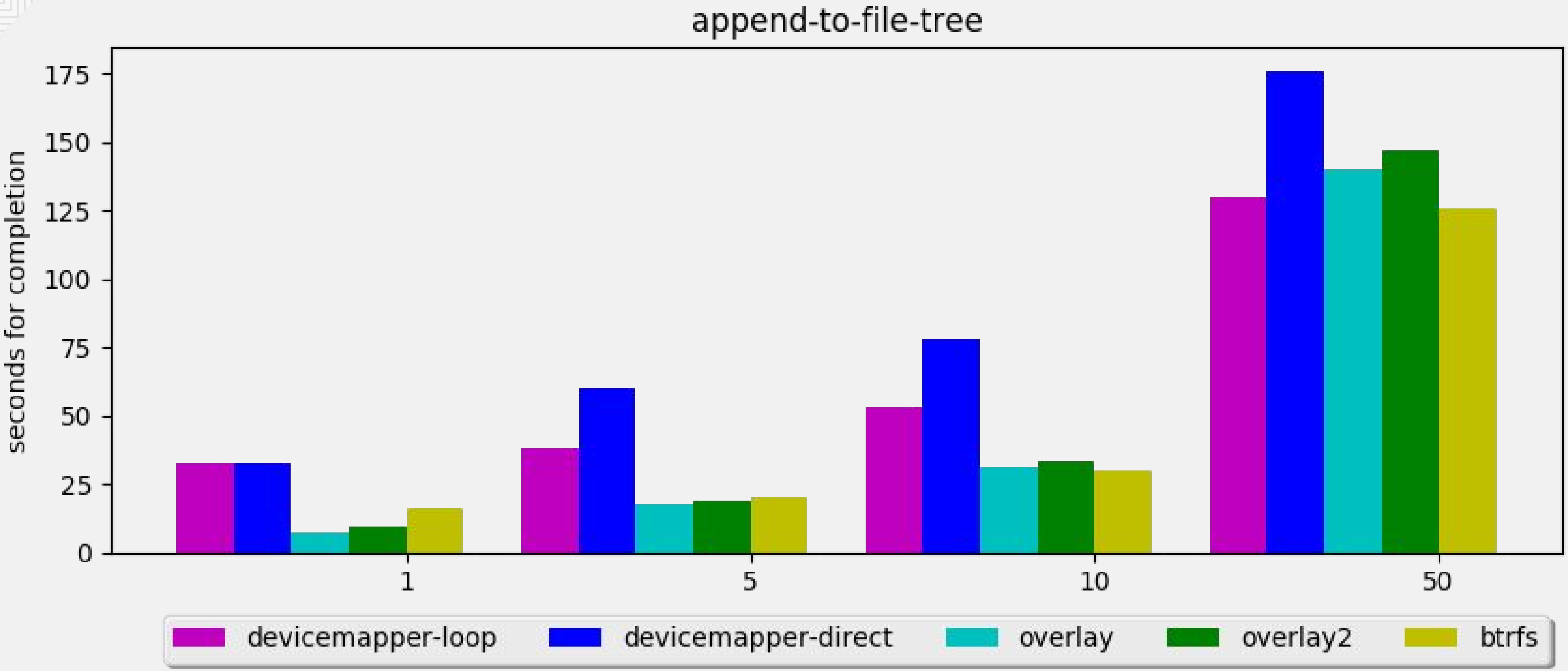
- Appending to Small Files
- Appending to Large Files
- Appending to File Tree



Naive benchmarking, for discussion purposes only. Don’t trust this!



Naive benchmarking, for discussion purposes only. Don’t trust this!



Naive benchmarking, for discussion purposes only. Don’t trust this!

Storage use cases

Technology	Attributes	Good Use Case	Bad Use Case
AUFS	Stable, Production Ready, Good Memory Use		High Write Activity
Btrfs	Mainline Kernel		High Write Activity
Overlay	Stable, Good Memory Use, Mainline Kernel		Container Churn
Devicemapper (loop)	Stable, Mainline Kernel		Production, Performance
Devicemapper (direct-lvm)	Stable, Production Ready, Mainline Kernel		

Resources

Storage Drivers in Docker: A Deep Dive
https://integratedcode.us/2016/08/30/storage-drivers-in-docker-a-deep-dive/

The Docker community has documented a good bit of this detail in the official storage driver documentation
https://docs.docker.com/engine/userguide/storagedriver/selectadriver/

Docker Issues and Tips (aufs/overlay/btrfs..)
https://github.com/AkihiroSuda/issues-docker#docker-issues-and-tips-aufsoverlaybtrfs

Comprehensive Overview of Storage Scalability in Docker (2014)
https://developers.redhat.com/blog/2014/09/30/overview-storage-scalability-docker/

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos