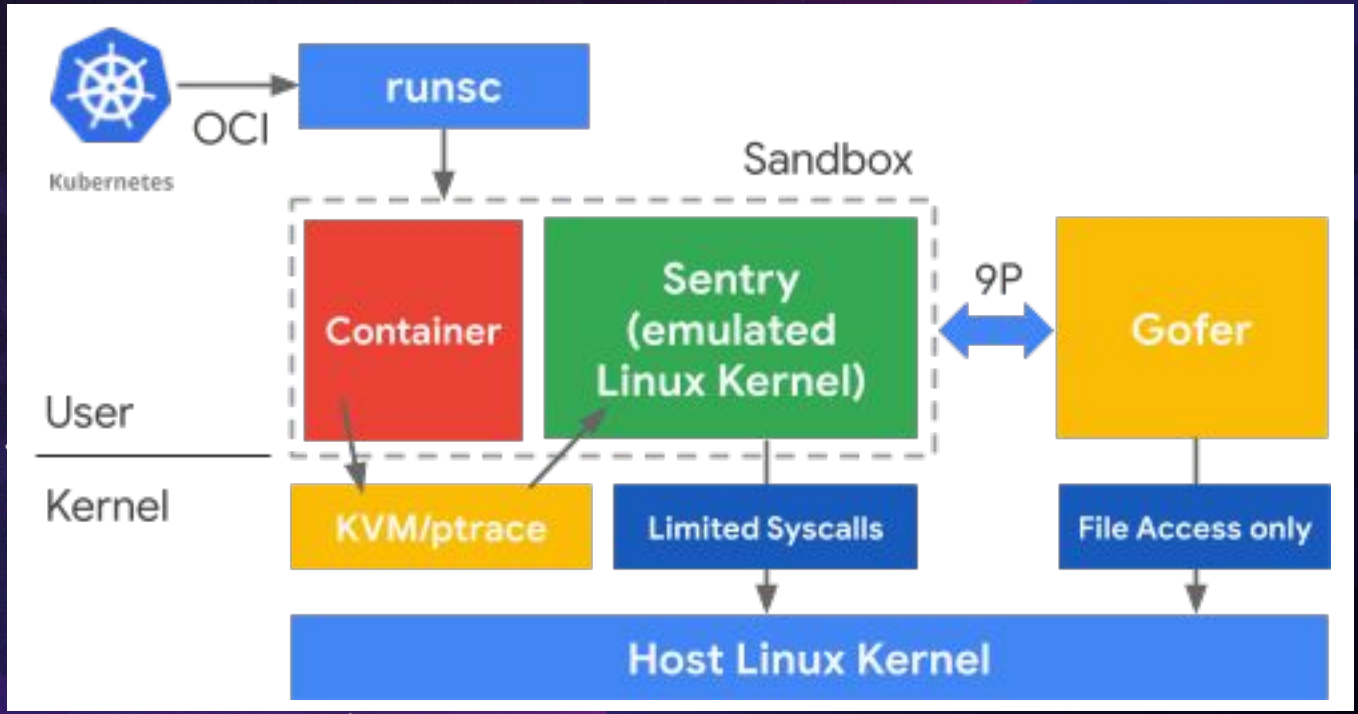
gVisor

Sam ✦ Jake ✦ Jack ✦ Will ✦
Jon

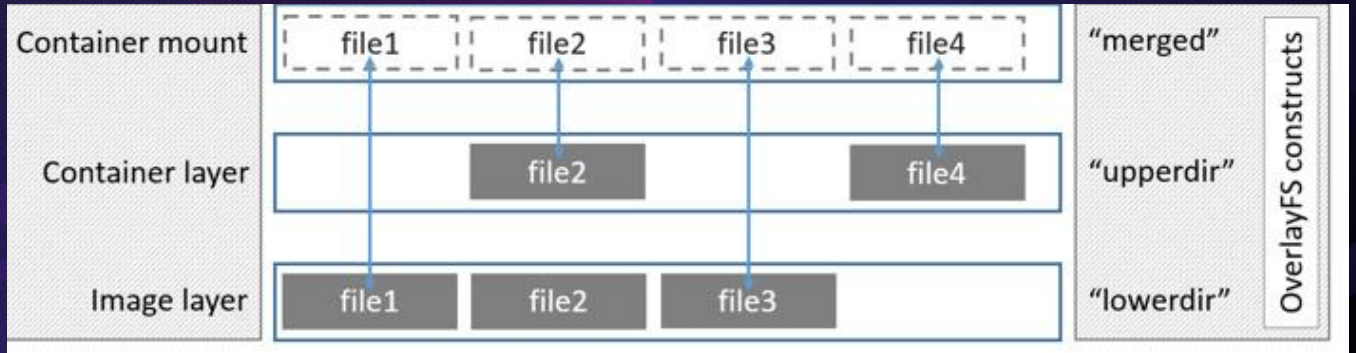
Introduction



Modules: Gofer - Filesystem Isolation

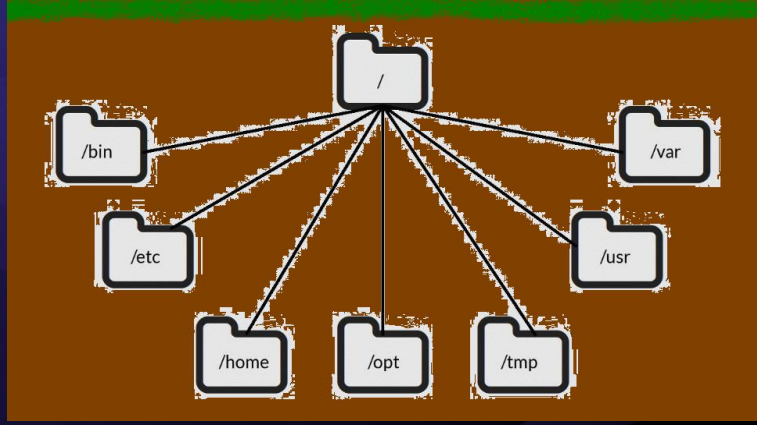
Gofer is a file proxy that allows gVisor sandboxed containers to access the file system

- Mount an overlayfs with an OS image and the container's code
- Make this mount the root of Gofer's filesystem namespace
- Whitelist process capabilities and allowed system calls with Seccomp+BPF



Modules: Gofer - 9P File Server

- Each container has a number of FDs to communicate with Gofer using the 9P protocol.
- Any attempt to open, read from, or modify anything in the filesystem is handled by Gofer
- A Goroutine is dispatched for each 9P connection and can spawn additional Goroutines to handle the next request.
- Non-blocking and blocking request handling



Abstractions: Threads and Processes

- gVisor sandbox appears as a single process to the host system
- Sentry creates a Task struct for each thread of execution.
- Tasks are dispatched as a goroutine.
  - Many-to-one user-space thread model provided by the Go language and can be
- Tasks are bundled together into a `TaskSet` for multithreaded applications.
- Tasks are scheduled by the Sentry and unknown to the host.
- Sentry can create host threads as needed to support Tasks.

Abstractions: Files

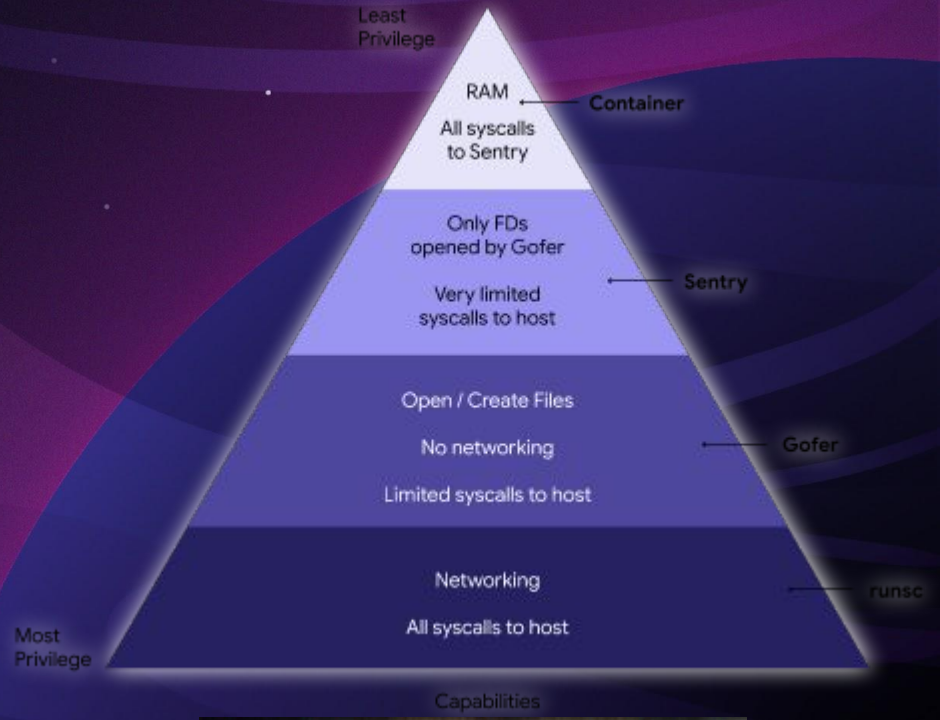
- Sentry asks Gofer for file descriptors to access host files.
- Can also be mapped into sandbox memory with gVisor's Mappable interface.
- Creates tmpfs filesystems internal to the sandbox for /tmp and /dev/shm
- Filesystem in gVisor consists of a tree of reference-counted Dentry nodes.
  - Dentry nodes are not associated with inodes.
  - Dentry nodes reference DentryImpl object that provides management information.

Abstractions: Memory

- All memory within a gVisor sandbox is managed by the Sentry using demand-paging and backed by a single memfs.
- Physical memory is handled entirely by the host.
- Sentry populates mappings from the host and allows the host to control demand-paging.
- Sentry will not demand an individual page of memory.
  - Instead, it uses memory-allocation heuristics to select regions.
- Sentry immediately releases freed memory back to the host.

Security

- Re-routing and intercepting system calls from the untrusted process.
- Individual implementations of Linux system calls in the Sentry.
- Reduced set of whitelisted system calls allowed for the Sentry.
- File operations provided by Gofer over the 9P protocol.



Performance (Compared to native Linux and runc)

- Container startup/tear down:
  - About *13% decrease* in performance compared to runc
- System call throughput:
  - In best case scenario: about *2.8x slower*
- Memory allocation:
  - Achieves about *40% the throughput* of native Linux
- File system access:
  - *12x slower* to access internal tmpfs in Sentry
  - *216x slower* to access external tmpfs through Gofer
- Networking:
  - Can keep up with native Linux for small file sizes (1MB)
  - Fails to scale well:
    - By 1GB, gVisor achieves *about half* the throughput of native Linux