K9P: Kubernetes as 9P Files

Terin Stock
20 November 2019

Terin Stock

@terinjokes

Kubernetes Engineer,
Cloudflare

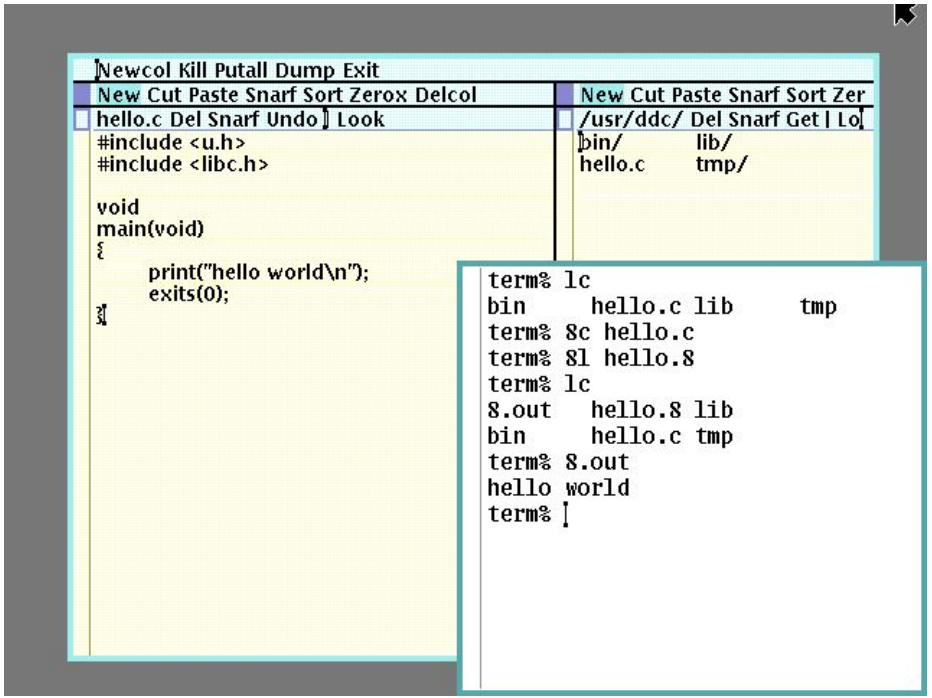
What if…

- Orchestrate compute across multiple hosts
- Orchestrate storage across multiple hosts
- Compute separate from users
- Applications running in a sandbox
- Service discovery

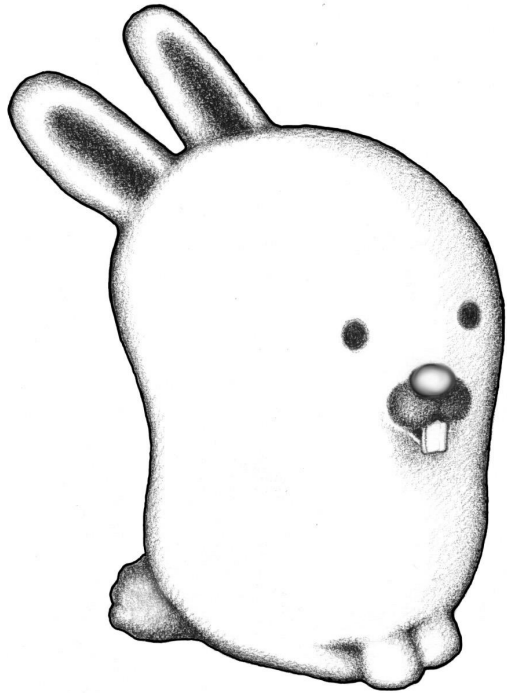
What if… it's 1992?

- Linux 1.0: +2 years
- Process Containers (later, cgroups): +14 years
- LMCTFY and Docker: +21 years
- Kubernetes: +23 years

What if… it's 1992? Plan 9!



Дмитрий Царьков (CC-SA-4.0)



Glenda, the Plan 9 Bunny

Drawn by Renée French

Plan 9 from Bell Labs

Rob Pike
Dave Presotto
Sean Dorward
Bob Flandrena
Ken Thompson
Howard Trickey
Phil Winterbottom

Bell Laboratories
Murray Hill, New Jersey 07974
USA

Motivation

By the mid 1980's, the trend in computing was away from large centralized time-shared computers towards networks of smaller, personal machines, typically UNIX 'workstations'. People had grown weary of overloaded, bureaucratic timesharing machines and were eager to move to small, self-maintained systems, even if that meant a net loss in computing power. As microcomputers became faster, even that loss was recovered, and this style of computing remains popular today.

In the rush to personal workstations, though, some of their weaknesses were over-

K9P

What is K9P?

- Kubernetes exposed as a 9P filesystem
- Goal to provide easy interaction with Kubernetes
  - Use familiar tools with clusters
  - Integrate into file-based workflows

9P…?

- Networked filesystem connecting Plan 9 systems and components
  - windows
  - network connections
  - processes
  - files

9P RPC Messages

- version
- auth
- attach
- flush
- walk
- create
- open
- read
- write
- clunk
- remove
- stat
- wstat
- error

9P in the modern era

- DataKit and VPNKit by Docker
- VirtFS in QEMU
- WSL by Microsoft
- Crostini and gVisor by Google

Linux Support

- Linux has native support when compiled with `CONFIG_NET_9P`.

```
# mount -t 9p -o trans=tcp,port=1564 198.51.100.1 /mnt/k8s
```

macOS and Windows

- Both operating systems have native implementations, not (currently) easily exposed to users.
  - macOS: 9P for VirtFS landed in 10.14.4
  - Windows: 9P is used for WSL
- In the meantime, there are FUSE implementations

What can I use K9P for?

Everything is a file!

Everything is a file!

```
$ ls -lp ./namespaces/kube-system/deployments/coredns
-rw-rw-rw- 1 4294967294 4294967294 5297 Jul  3 17:03 data.yaml
-rw-rw-rw- 1 4294967294 4294967294 0    Jul  3 17:03 scale
-r--r--r-- 1 4294967294 4294967294 0    Jul  3 17:03 status
```

Pipelines and Integrations

Pipelines and Integrations

Scale by writing to a file

```
$ echo 5 > ./namespaces/kube-system/deployments/coredns/scale
```

Pipelines and Integrations

Search for non-running Pods

```
$ grep -L Running ./namespaces/kube-system/pods/*/status

./namespaces/kube-system/pods/coredns-5fc75767bb-7f9gx/status
```

Pipelines and Integrations

Tail Pod logs

```
$ tail ./namespaces/kube-system/pods/coredns-*/logs/*

2019-07-04T00:20:08.602Z [ERROR] plugin/errors: 2
www.example.com. A: read udp
198.51.100.34:58230->198.51.100.2:53: i/o timeout
```

Pipelines and Integrations

Modify cluster resources

```
$ sed -i 's/1.6.4/1.6.5/'
./namespaces/kube-system/deployments/coredns/data.yaml
```

Controllers and Operators

Controllers and Operators

- No language libraries required
  - Everything is a file.
  - Most programming languages have great filesystem support.

Controllers and Operators

- No networking code.
- No network access.
  - 9P client can run at a different level of network access

Controllers and Operators

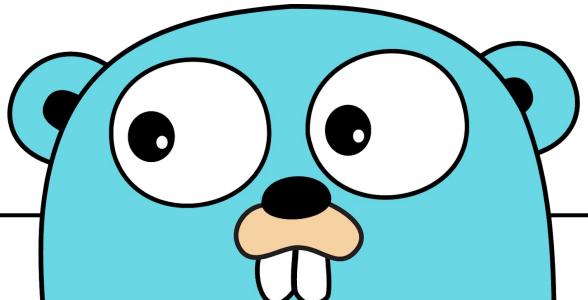
- Very testable
  - Inputs and outputs are just files.

Demo

Implementing K9P

Implementing K9P

- Server implemented in Go
  - 9P server exposed over TCP
  - Uses Kubernetes's client-go
- Each mount is associated with a new connection to Kubernetes



Drawn by Renée French

9P and Go

- Many available implantations: p9p, neinp, styx, p9 (gVisor)
  - Each has a different level of implementation, abstraction, and paradigms
  - K9P uses p9p from Docker.

Files

- Kubernetes resources are mapped to directory layouts
  - Cluster-scope resources are mapped to `/cluster`
  - Namespace-scope resources are mapped inside of `/namespaces`
- Each supported resource type has a handler implementation.
  - Would like to derive implementations where possible.

Challenges

Polling

- 9P has no subscription mechanism
  - No way for the server to push new information to clients
  - Required to open directories and files to observe changes

Managing State

- This is my first virtual filesystem, continue to be unsure about the best way to implement and manage the filesystem state.
  - Currently, fids are mapped to resources and a lot of state is generated on demand. Performance isn't often great.

Versions

- The Kubernetes API uses resources versions to prevent modifications based on old versions of resources.
- User's view of a file may fall out of sync with the server's resource version.
  - Future improvement: associate a qid with the resource's version, know when a conflict might happen.

Conflicts and Errors

- Inflexibility to report conflicts and issues
  - Can only return errors to client requests
  - Client often attempt to turn stringy errors into standard system errors.

Authentication

- 9P defers authentication to filesystem implementations

> The 9P protocol does not prescribe an authentication method. Instead, client and server communicate by reading from and writing to a special file.

- Future improvement to setup an authentication protocol to exchange user or session tokens.

Discussion

9P and more

- Only scratched the surface of what 9P can do.
  - Go out and experiment
  - Build your own filesystems
    - KubeCon talk videos as a filesystem? It could be the next YouTube!
- NBD implements network-backed block devices.
- CUSE allows character devices to be implemented in userspace.

Wide Area SHell

- Puppet's wash is a similar idea
  - Shell to interact with remote and cloud-native infrastructure with unix-like tools
  - Some support as a FUSE filesystem

9P's Future?

- Strings for UIDs and GIDs didn't catch on elsewhere.
- Lacks support for some expected things:
  - symlinks can be iffy
  - xattrs
  - sockets
- Protocol can be versioned and extended, but who's coordinating?

Use K9P at your own risk!

It might delete something important.

More K9P

- Learn more + code: https://terin.ee/u/k9p
- Find me in the hallway track for questions.

Back Matter

Colophon

The main body text, including headers, were set in Cambo by Argentinian foundry Huerta Tipográfica based on the style of traditional Khmer type.

Monospace text was set in Anonymous Pro by Minnesota font designer Mark Simonson. It was inspired by mid-90s freeware Macintosh font Anonymous 9.

The one quote was set in Open Sans, a humanist font designed by Coloradan Steve Matteson at the Monotype foundry.

Colophon

Glenda, the Plan 9 Bunny; the Go Gopher; and the unnamed dachshund were designed by Renée French.

Glenda, the Plan 9 Bunny is copyright Alcatel-Lucent S.A., used with permission.

The Go Gopher is licensed under the Creative Commons 3.0 Attributions license.

The dachshund was shamelessly lifted from the cover of *The Practice of Programming*, copyright Alcatel-Lucent S.A.