Network Shells

Michael Smith

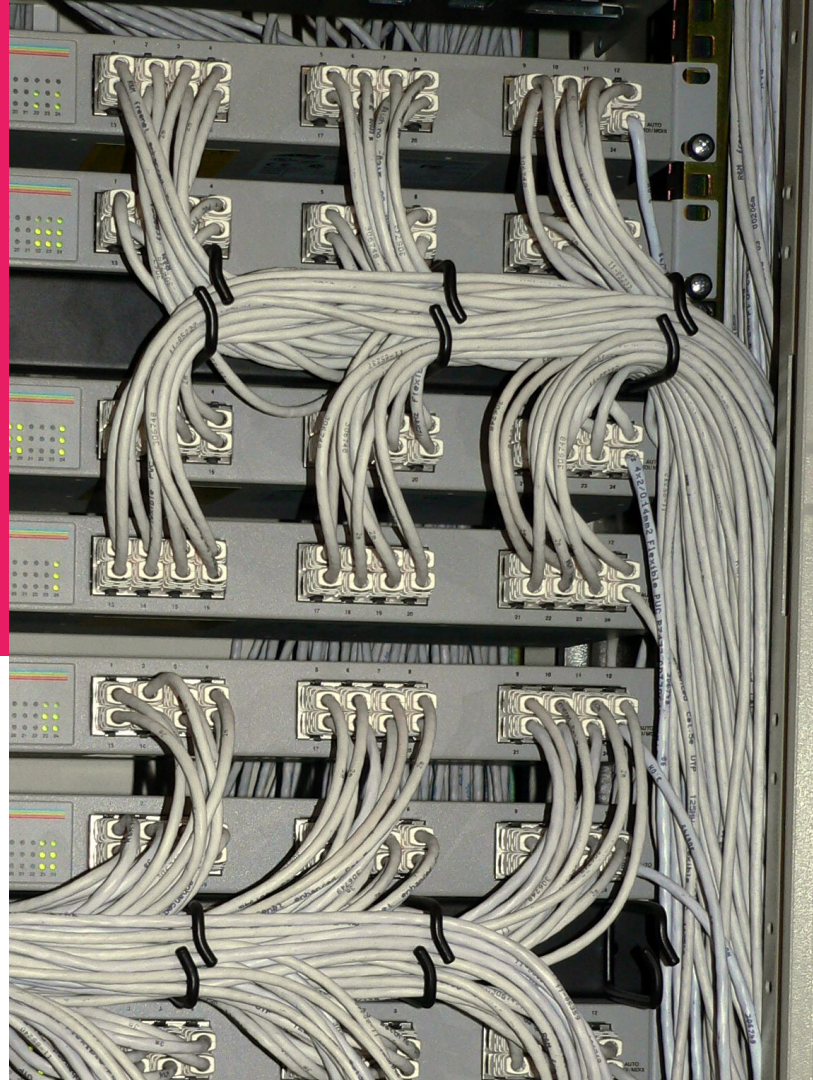


Image: https://commons.wikimedia.org/wiki/File:Network-connections.png

```
chealer@vinci:/usr/share/doc/bash$ export LC_ALL=C
chealer@vinci:/usr/share/doc/bash$ cd ~chealer/
chealer@vinci:~$ ls
Cloutier  Ido         Musique  logs      skolo  sources
Desktop   Mes images  boston   ncix.png  smb4k  vieux
chealer@vinci:~$ #Why is there color when calling ls without arguments?
chealer@vinci:~$ which ls
/bin/ls
chealer@vinci:~$ $(!!)
$(which ls)
Cloutier  Ido         Musique  logs      skolo  sources
Desktop   Mes images  boston   ncix.png  smb4k  vieux
chealer@vinci:~$ type ls #"ls" doesn't just run /bin/ls
ls is aliased to `ls --color=auto'
chealer@vinci:~$ echo $PS1
${debian_chroot:+($debian_chroot)}\u@\h:\w\$
chealer@vinci:~$ sh
sh-3.1$ echo $PS1
\s-\v\$
sh-3.1$ echo $BASH_VERSION
3.1.17(1)-release
sh-3.1$ ls
Cloutier  Ido         Musique  logs      skolo  sources
Desktop   Mes images  boston   ncix.png  smb4k  vieux
sh-3.1$ echo $SHELLOPTS # ls isn't an alias in POSIX mode
braceexpand:emacs:hashall:histexpand:history:interactive-comments:monitor:posix
sh-3.1$ kill
kill: usage: kill [-s sigspec | -n signum | -sigspec] pid | jobspec ... or kill
-l [sigspec]
sh-3.1$ /bin/kill &> killerror # collect stdout and stderr of $ /bin/kill; in ki
llerror
sh-3.1$ wc -l !$
wc -l killerror
7 killerror
sh-3.1$ type kill # kill doesn't just run /bin/kill, even in POSIX mode.
kill is a shell builtin
sh-3.1$ !$ -n 9 $$ # OK, kill self
kill -n 9 $$ # OK, kill self
Killed
chealer@vinci:~$
```

What does a Shell give us?

- A REPL
- Repeatability
- Direct access to system operations
- User-focused design
- Hierarchical context & sense of place

Image: https://upload.wikimedia.org/wikipedia/commons/8/84/Bash_demo.png

```
chealer@vinci:/usr/share/doc/bash$ export LC_ALL=C
chealer@vinci:/usr/share/doc/bash$ cd ~chealer/
chealer@vinci:~$ ls
Cloutier  Ido         Musique  logs      skolo  sources
Desktop   Mes images  boston   ncix.png  smb4k  vieux
chealer@vinci:~$ #Why is there color when calling ls without arguments?
chealer@vinci:~$ which ls
/bin/ls
chealer@vinci:~$ $(!!)
$(which ls)
Cloutier  Ido         Musique  logs      skolo  sources
Desktop   Mes images  boston   ncix.png  smb4k  vieux
chealer@vinci:~$ type ls #"ls" doesn't just run /bin/ls
ls is aliased to `ls --color=auto'
chealer@vinci:~$ echo $PS1
${debian_chroot:+($debian_chroot)}\u@\h:\w\$
chealer@vinci:~$ sh
sh-3.1$ echo $PS1
\s-\v\$
sh-3.1$ echo $BASH_VERSION
3.1.17(1)-release
sh-3.1$ ls
Cloutier  Ido         Musique  logs      skolo  sources
Desktop   Mes images  boston   ncix.png  smb4k  vieux
sh-3.1$ echo $SHELLOPTS # ls isn't an alias in POSIX mode
braceexpand:emacs:hashall:histexpand:history:interactive-comments:monitor:posix
sh-3.1$ kill
kill: usage: kill [-s sigspec | -n signum | -sigspec] pid | jobspec ... or kill
-l [sigspec]
sh-3.1$ /bin/kill &> killerror # collect stdout and stderr of $ /bin/kill; in ki
llerror
sh-3.1$ wc -l !$
wc -l killerror
7 killerror
sh-3.1$ type kill # kill doesn't just run /bin/kill, even in POSIX mode.
kill is a shell builtin
sh-3.1$ !$ -n 9 $$ # OK, kill self
kill -n 9 $$ # OK, kill self
Killed
chealer@vinci:~$
```

What does a Shell give us?

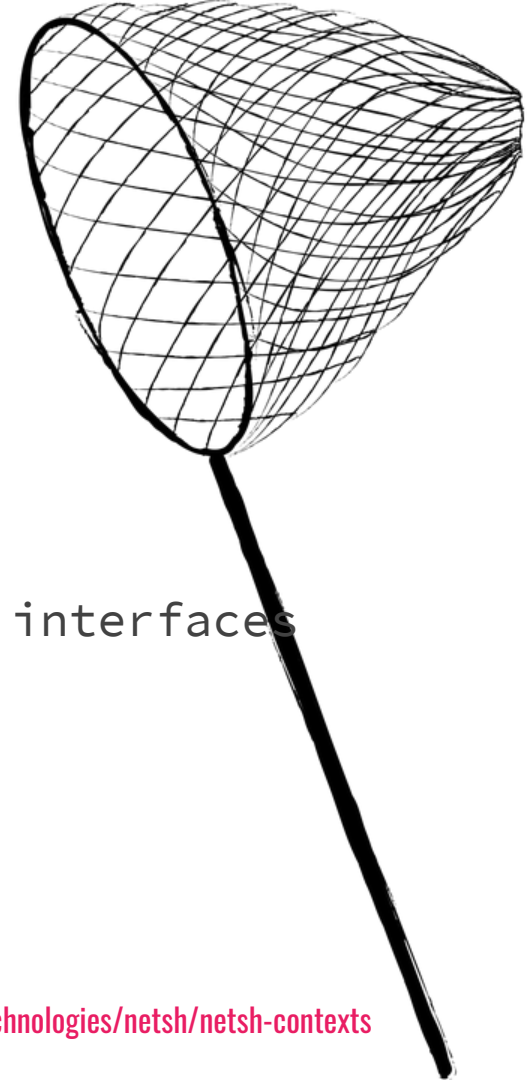
- A REPL
- Repeatability
- Direct access to system operations
- User-focused design
- Hierarchical context & sense of place

Image: https://upload.wikimedia.org/wikipedia/commons/8/84/Bash_demo.png

Management at a distance (netsh)

```
Netsh: Configure DHCP servers with
    netsh -r RemoteMachine -u domain\username
    [RemoteMachine] netsh>interface
    [RemoteMachine] netsh interface>ipv6
    [RemoteMachine] netsh interface ipv6>show interfaces
```





Reference: https://docs.microsoft.com/en-us/windows-server/networking/technologies/netsh/netsh-contexts

Management at a distance (netsh)

```
Netsh: Configure DHCP servers with
    netsh Location moteMachine -u domain\username
    [RemoteMachine] nets Hierarchical context ace
    [RemoteMachine] netsh interface>ipv6
    [RemoteMachine] netsh interface ipv6>show interfaces
```

Simpler commands

Reference: https://docs.microsoft.com/en-us/windows-server/networking/technologies/netsh/netsh-contexts

Management at a distance (WSMan)

WSMan (in Powershell): Manage Windows remotely with

```
Set-Location -Path WSMan:\SERVER01
Get-ChildItem -Path .
Set-Item Client\TrustedHosts *.domain2.com -Concatenate
```

Reference: https://docs.microsoft.com/en-us/powershell/module/microsoft.wsman.management/about/about_wsman_provider

Management at a distance (other examples)

---

github.com/fishi0x01/vsh - for manipulating Vault secrets

code.vmware.com/web/tool/11.4.0/vmware-powercli - for managing everything VMWare in PowerShell

github.com/vmware/rvc - a console for vSphere

metacpan.org/pod/distribution/ldapsh/ldapsh - a shell for editing LDAP

Interlude: PowerShell Providers

```
Get-Item alias:
```

```
Get-ChildItem
HKLM:\SOFTWARE\
```

```
Set-Location cert:
```

```
Get-ChildItem
```

Reference:
https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_providers

Provider	Drive	Data store
Alias	Alias:	PowerShell aliases
Certificate	Cert:	x509 certificates for digital signatures
Environment	Env:	Windows environment variables
FileSystem	(*)	File system drives, directories, and files
Function	Function:	PowerShell functions
Registry	HKLM:, HKCU:	Windows registry
Variable	Variable:	PowerShell variables
WSMan	WSMan:	WS-Management configuration information

Hide the network (NFS, SSHFS, GlusterFS)

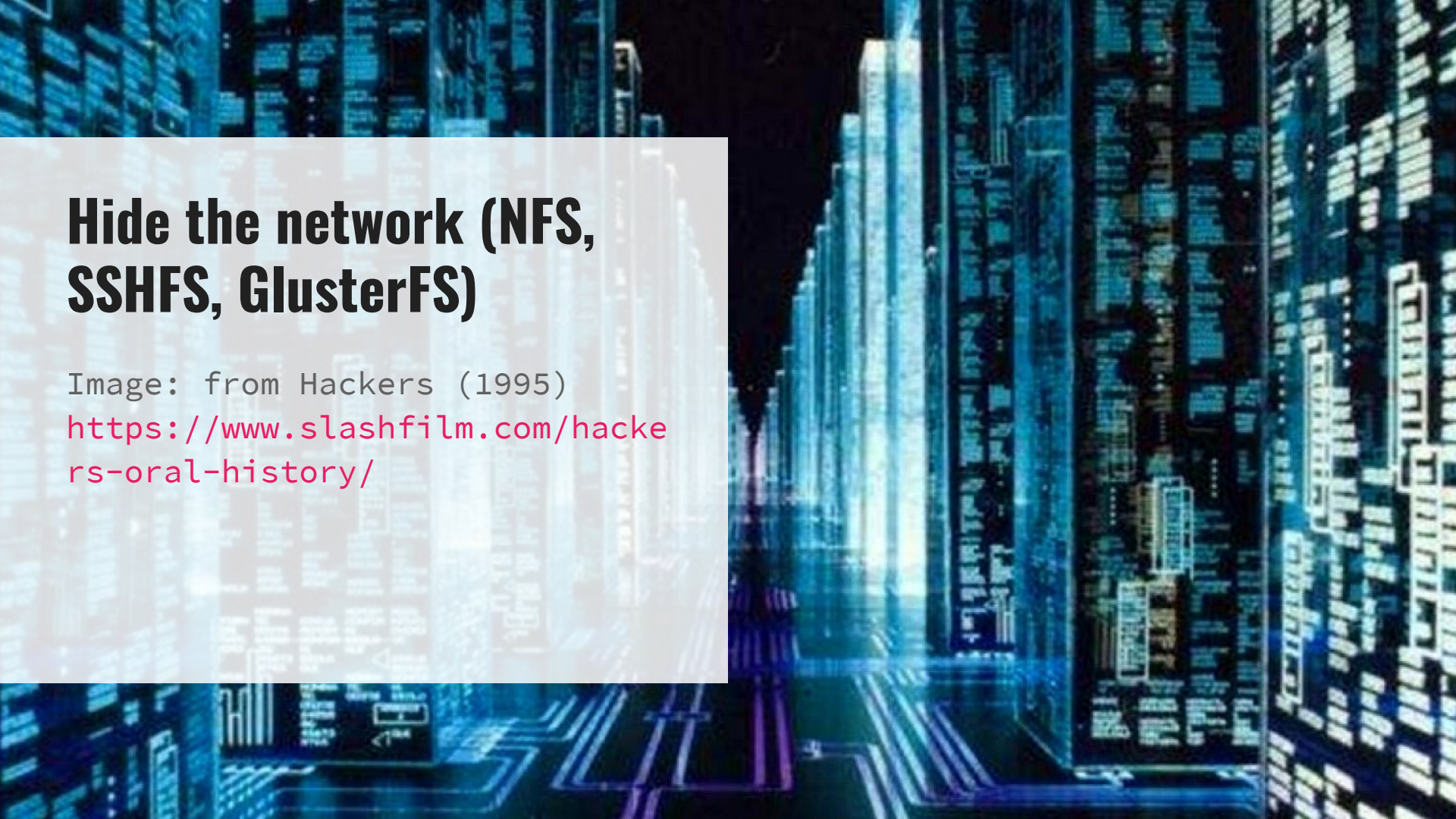
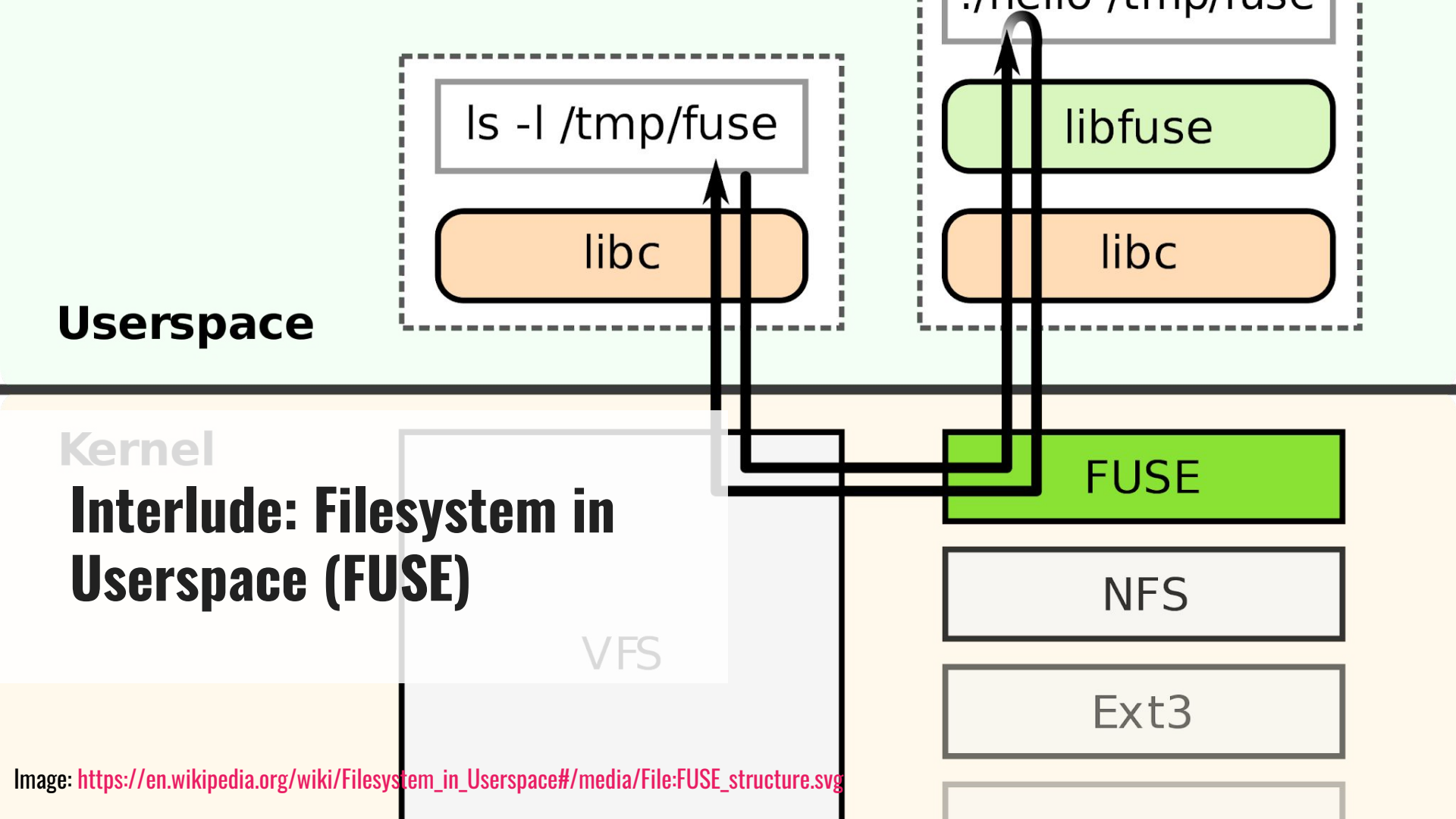


Image: from Hackers (1995)
https://www.slashfilm.com/hackers-oral-history/



Interlude: Filesystem in Userspace (FUSE)

Image: https://en.wikipedia.org/wiki/Filesystem_in_Userspace#/media/File:FUSE_structure.svg

Remote execution in your shell (NSH by BMC)

```
nsh @% cd //host1
nsh host1% uname -a
host1
nsh host1% tail /var/log/messages
Oct 24 17:21:43 localhost liblogging-stdlog:  [origin
software="rsyslogd" swVersion="8.24.0" x-pid="486"
x-info="http://www.rsyslog.com"] start
...
```

Reference: https://docs.bmc.com/docs/ServerAutomation/86/developing/working-with-network-shell/nsh-cheat-sheet

Remote execution in your shell (NSH by BMC)

```
nsh @% cd //host1


nsh host1% tail /var/log/messages


nsh host1% cp //host2/home/myuser/some_archive.tar /tmp/
```

Reference: https://docs.bmc.com/docs/ServerAutomation/86/developing/working-with-network-shell/nsh-cheat-sheet

Remote execution in your shell (NSH by BMC)

```
nsh host1% cp //host2/home/myuser/some_archive.tar /tmp/
```

```
nsh host1% cp /tmp/some_archive.tar //@/some/dir
```

```
nsh @% ndf; nps; ntop; nmem; nexec
```

Reference: https://docs.bmc.com/docs/ServerAutomation/86/developing/working-with-network-shell/nsh-cheat-sheet

Remote execution in your shell (Wash)

```
wash . > ls
aws/
docker/
gcp/
kubernetes/
```

```
wash . > ls gcp/Wash/compute
michael-test-instance

wash . > tree gcp/Wash/storage
gcp/Wash/storage
└── some-wash-stuff
    ├── an\ example\ folder
    │   └── static.sh
    └── reaper.sh
```

Reference: https://puppetlabs.github.io/wash/

Remote execution in your shell (Wash)

```
wash gcp/Wash/compute/a-host > tail fs/var/log/messages
Oct 24 17:21:43 localhost liblogging-stdlog:  [origin
software="rsyslogd" swVersion="8.24.0" x-pid="486"
x-info="http://www.rsyslog.com"] start
…
wash gcp/Wash/compute/a-host > meta .
cpuPlatform: Intel Broadwell
creationTimestamp: "2019-10-24T10:21:33.167-07:00"
...
```

Reference: https://puppetlabs.github.io/wash/

Remote execution in your shell (Wash)

```
wash . > find . -crtime -30d -action exec
./aws/eso/resources/ec2/instances/wash-test_i-0c64807d0ef65c585
./gcp/Wash/compute/a-host
./kubernetes/gke_shared-k8s_us-west1-a_shared-k8s-dev/monitoring/pods/prometheus-64c6dfb964-tzmbr/prometheus
...
wash aws/eso > find resources -meta '.Tags[?]' .Key Team -a .Value wash
./ec2/instances/wash-test_i-0c64807d0ef65c585
...
```

Reference: https://puppetlabs.github.io/wash/

Why do we love them?

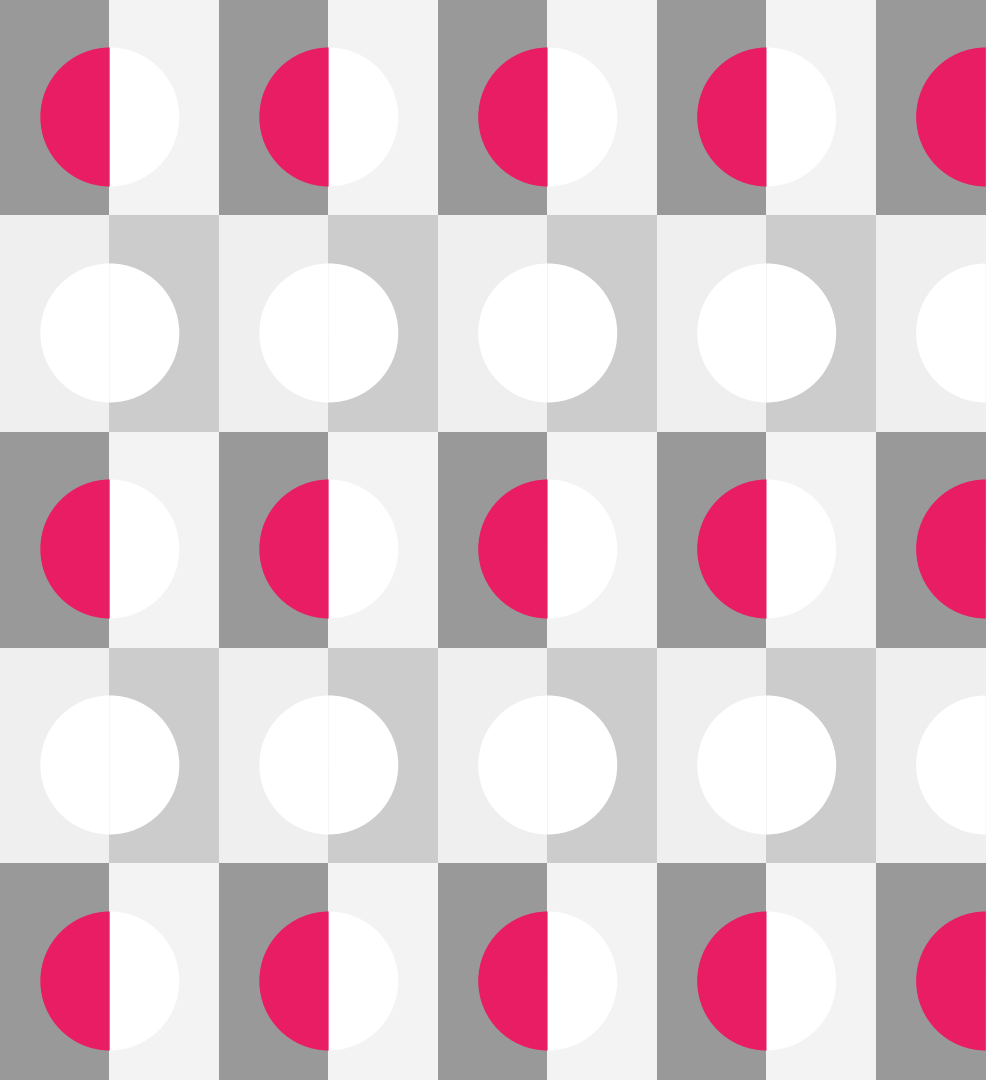
Why do we love them?

Reinventing the wheel

Caching

Cross-platform clients

Network resources are often not files



Thank You