



VIDEO

**ABSTRACT**

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.

SLIDES

The slide features a red and purple abstract background with circular patterns. At the top left is the Open Source Summit logo. In the center, the title "Classic Tools for the Modern Age: Introducing Wash, The Cloud Native Shell" is displayed in large, bold, white font. Below the title, the speaker's name "Michael Smith / Puppet, Inc." is shown in a smaller white box. At the bottom left, there are hashtags "#ossummit" and "@lasthemy". In the bottom right corner, the "THE LINUX FOUNDATION" logo is visible.

Classic Tools for the Modern Age: Introducing Wash, The Cloud Native Shell

Michael Smith / Puppet, Inc.

#ossummit @lasthemy

THE LINUX FOUNDATION





VIDEO



ABSTRACT

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.

SLIDES

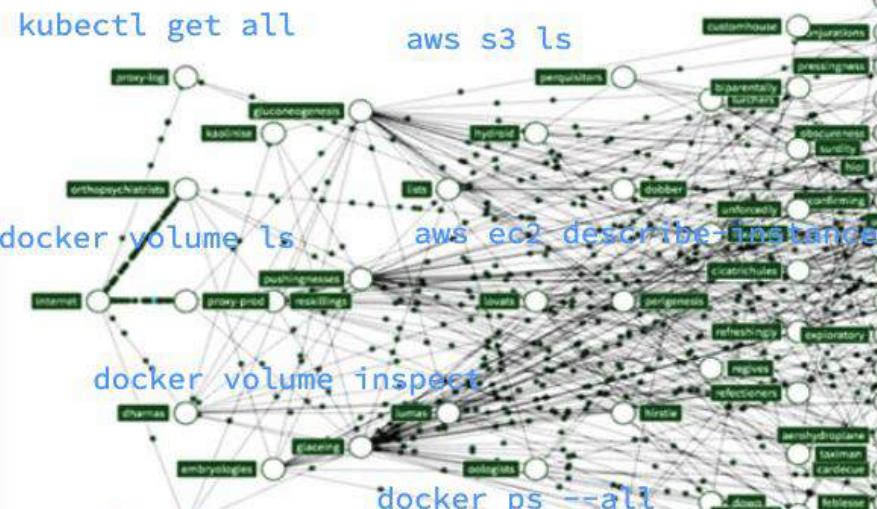


Image: www.honeycomb.io/microservices





VIDEO



ABSTRACT

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.

SLIDES



`kubectl logs -f container1`

`ssh tail -f /var/log/messages`

`docker exec container1 tail -f /data/logs`

`aws cloudtrail ...?`



VIDEO



SLIDES

`kubectl exec``docker exec``ssh``winrm`

4

ABSTRACT

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.





VIDEO



ABSTRACT

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.

SLIDES

`docker stop | docker stack rm``kubectl delete``aws delete-?``gcloud compute instances delete`

5



VIDEO

**ABSTRACT**

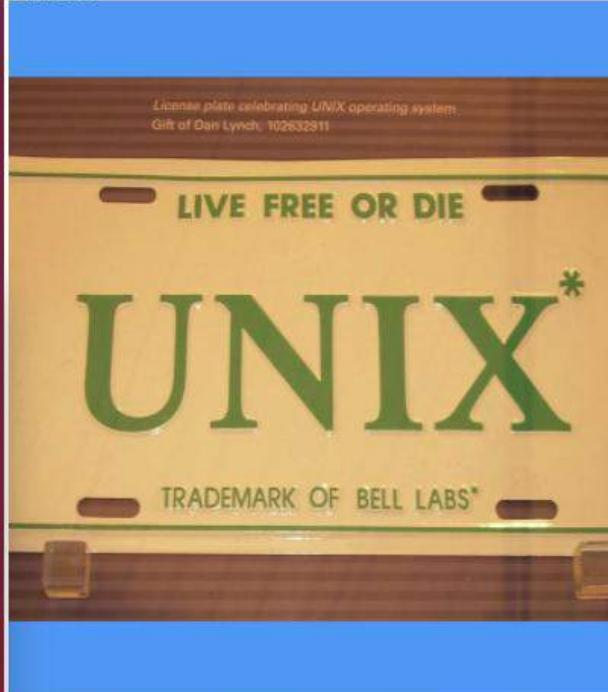
Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.

SLIDES

`ls -l``tail -f``exec``rm`

6





VIDEO



SLIDES

What Could We Build?

`ps/top``aggregate logs``cleanup unused resources`

7

ABSTRACT

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.





VIDEO



SLIDES



wash

<https://puppetlabs.github.io/wash>

8

ABSTRACT

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.



SCREEN SHARE

```
wash
Welcome to Wash! Try 'docs .'
wash . > docs .
Welcome to Wash, a UNIX-like shell that lets you manage all your entries as if
they were files and directories. This entry represents the Wash root. 'ls'-ing
it yields all the configured plugins.

SUPPORTED ACTIONS
* list
  - ls .
    Type 'docs <child>' to view an ls'ed child's documentation
  - cd .
    Use 'cd' to return to Wash's starting directory
  - stree .
    Gives you a high-level overview of the kinds of things that you will
    encounter when you 'cd' and 'ls' through this entry
  - (anything else that works with directories [e.g. 'tree'])

Type 'help' to see a list of all the built-in Wash commands.
wash . > ls
aws/
docker/
gcp/
kubernetes/
wash . >
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)
variable or $HOME/.aws/config to find profiles and configure the SDK. See
https://docs.aws.amazon.com/sdk-for-php/v3/developer-guide/guide\_credentials\_profiles.html
for more details on how to setup AWS profiles.

You can limit the loaded profiles by adding

aws:
  profiles: [profile_1, profile_2]

to Wash's config file.

The AWS plugin currently supports EC2 and S3. IAM roles are supported when configured
as described here. Note that currently region will also need to be specified with the
profile.

If using MFA, Wash will prompt for it on standard input. Credentials are valid for 1 hour.
They are cached under wash/aws-credentials in your user cache directory so they can be
re-used across server restarts. Wash may have to re-prompt for a new MFA token in response
to navigating the Wash environment to authorize a new session.

SUPPORTED ACTIONS
* list
  - ls aws
    Type 'docs <child>' to view an ls'ed child's documentation
  - cd aws
    Use 'cd' to return to Wash's starting directory
  - stree aws
    Gives you a high-level overview of the kinds of things that you will
    encounter when you 'cd' and 'ls' through this entry
  - (anything else that works with directories [e.g. 'tree'])

Type 'help' to see a list of all the built-in Wash commands.
wash . > stree
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)
wash . > stree
.
+-- kubernetes
    +-- [context]
        +-- [namespace]
            +-- pods
                +-- [pod]
                    +-- [container]
                        +-- log
                        +-- metadata.json
                        +-- fs
                            +-- [dir]
                                +-- [dir]
                                    +-- [file]
                                    +-- [file]
            +-- persistentvolumeclaims
                +-- [persistentvolumeclaim]
                    +-- [dir]
                        +-- [dir]
                            +-- [file]
                            +-- [file]
                    +-- [file]
                    +-- [file]
.
+-- docker
    +-- containers
        +-- [container]
            +-- log
            +-- metadata.json
            +-- fs
                +-- [dir]
                    +-- [dir]
                        +-- [file]
                        +-- [file]
    +-- volumes
        +-- [volume]
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)

gcp [project]
  compute
    instance
      console.out
      metadata.json
      fs
        [dir]
          [dir]
            [file]
        [file]
    storage
      [bucket]
        [prefix]
          [prefix]
          [object]
        [object]
    firestore
      [collection]
        [document]
          data.json
        [collection]
    pubsub
      [topic]
    cloud_functions
      [cloud_function]
        log
    cloud_run
      [service]
        log
aws
  [profile]
    resources
```



SCREEN SHARE

```
wash . > Default (zsh)
.
+-- firestore
|   +-- [collection]
|   |   +-- [document]
|   |   |   +-- data.json
|   |   |   +-- [collection]
|   +-- pubsub
|       +-- [topic]
|   +-- cloud_functions
|       +-- [cloud_function]
|           +-- log
|   +-- cloud_run
|       +-- [service]
|           +-- log
.
+-- aws
    +-- [profile]
        +-- resources
            +-- s3
                +-- [bucket]
                    +-- [prefix]
                        +-- [prefix]
                            +-- [object]
                            +-- [object]
            +-- ec2
                +-- instances
                    +-- [instance]
                        +-- [console.out]
                        +-- metadata.json
                        +-- fs
                            +-- [dir]
                                +-- [dir]
                                    +-- [file]
                                    +-- [file]
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)
└ ec2
  └ instances
    └ [instance]
      └ [console.out]
      └ metadata.json
      └ fs
        └ [dir]
          └ [dir]
            └ [file]
            └ [file]

wash . > tree -L 5 aws
aws
└ personal
  └ resources
    └ ec2
      └ instances
        └ i-080ae094d218db0aa
        └ i-0f721c2fe68bc082a
    └ s3
      └ puppet-wash
        └ MAINT.md
        └ README.md
        └ wash-logo-lg.png
        └ wash-logo.png

8 directories, 4 files
wash . > which tree
/usr/local/bin/tree
wash . > lsd -l gcp
dr-xr-x--- michaelsmith staff 0 B Thu Jun 18 09:18:59 2020 ⇨Cpp Projects
dr-xr-x--- michaelsmith staff 0 B Thu Jun 18 09:18:59 2020 ⇨platform-dujour-dev
dr-xr-x--- michaelsmith staff 0 B Thu Jun 18 09:18:59 2020 ⇨Wash
wash . >
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)
wash . > which tree
/usr/local/bin/tree
wash . > lsd -l gcp
dr-xr-x--- michaelsmith staff 0 B Thu Jun 18 09:18:59 2020 ⇔Cpp Projects
dr-xr-x--- michaelsmith staff 0 B Thu Jun 18 09:18:59 2020 ⇔platform-dujour-dev
dr-xr-x--- michaelsmith staff 0 B Thu Jun 18 09:18:59 2020 ⇔Wash
wash . > help
Type 'exit' to exit the shell at any time.

Usage:
[command]

Available Commands:
delete      Deletes the entries at the specified paths
docs        Displays the entry's documentation
find         Prints out all entries that satisfy the given expression
help         Help about any command
ls          Lists the children of the specified paths, or current directory if not specified
meta        Prints the metadata of the given entries
signal      Sends the specified signal to the entries at the specified paths
stree       Displays the entry's stree (schema-tree)
tail        Displays new output of files or resources with the stream action
wclear      Clears the cache at the specified paths, or current directory if not specified
wexec       Executes the given command on the indicated target
whistory    Prints the wash command history, or journal of a particular item
winfo       Wash's version of 'stat'. Prints the entries' info at the specified paths
wps         Lists the processes running on the indicated compute instances

Flags:
-h, --help   help for this command

Use " [command] --help" for more information about a command.
wash . > find . -meta .State.Name running
```



SCREEN SHARE

iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help

Default (zsh)

```
PrivateDnsName: ip-172-31-31-128.us-west-2.compute.internal
PrivateIpAddress: 172.31.31.128
ProductCodes: null
PublicDnsName: ec2-54-191-180-180.us-west-2.compute.amazonaws.com
PublicIpAddress: 54.191.180.180
RamdiskId: null
RootDeviceName: /dev/xvda
RootDeviceType: ebs
SecurityGroups:
- GroupId: sg-08aee5609a1c06572
 GroupName: launch-wizard-1
SourceDestCheck: true
SpotInstanceRequestId: null
SriovNetSupport: null
State:
  Code: 16
  Name: running
StateReason: null
StateTransitionReason: ""
SubnetId: subnet-f2c68696
Tags:
- Key: Role
  Value: Webserver
- Key: Org
  Value: Engineering
- Key: Team
  Value: Wash
VirtualizationType: hvm
VpcId: vpc-7dd58619
wash . > find . -meta .State.Name running
./aws/personal/resources/ec2/instances/i-080ae094d218db0aa
./aws/personal/resources/ec2/instances/i-0f721c2fe68bc082a
wash . >
```



SCREEN SHARE

The screenshot shows a Mac desktop with a dark-themed iTerm2 window open. The window title is "iTERT2" and the tab title is "Default (zsh)". The terminal output displays the results of a "wash" command run against local files. The output is organized into sections for each instance found:

```
VirtualizationType: hvm
VpcId: vpc-7dd58619
wash . > find . -meta .State.Name running
./aws/personal/resources/ec2/instances/i-080ae094d218db0aa
./aws/personal/resources/ec2/instances/i-0f721c2fe68bc082a
wash . > find . -meta .State.Name running | xargs winfo
./aws/personal/resources/ec2/instances/i-0f721c2fe68bc082a:
  Name: i-0f721c2fe68bc082a
  CName: i-0f721c2fe68bc082a
  Actions:
    - signal
    - list
    - exec
    - delete
  Attributes:
    crttime: 2020-06-17T20:14:19Z
    mtime: 2020-06-17T20:14:20Z
    os:
      login_shell: posixshell
./aws/personal/resources/ec2/instances/i-080ae094d218db0aa:
  Name: i-080ae094d218db0aa
  CName: i-080ae094d218db0aa
  Actions:
    - signal
    - list
    - exec
    - delete
  Attributes:
    crttime: 2020-06-17T20:14:19Z
    mtime: 2020-06-17T20:14:20Z
    os:
      login_shell: posixshell
wash . >
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)

- kind: compute#accessConfig
  name: External NAT
  natIP: 35.199.171.212
  networkTier: PREMIUM
  type: ONE_TO_ONE_NAT
  fingerprint: VxmZF_2YgX8=
kind: compute#networkInterface
name: nic0
network: https://www.googleapis.com/compute/v1/projects/wash-247718/global/networks/default
networkIP: 10.138.0.3
subnetwork: https://www.googleapis.com/compute/v1/projects/wash-247718/regions/us-west1/subnetworks/default
reservationAffinity:
  consumeReservationType: ANY_RESERVATION
scheduling:
  automaticRestart: true
  onHostMaintenance: MIGRATE
selfLink: https://www.googleapis.com/compute/v1/projects/wash-247718/zones/us-west1-a/instances/michael-test-instance
serviceAccounts:
- email: 668600244557-compute@developer.gserviceaccount.com
  scopes:
    - https://www.googleapis.com/auth/devstorage.read_only
    - https://www.googleapis.com/auth/logging.write
    - https://www.googleapis.com/auth/monitoring.write
    - https://www.googleapis.com/auth/servicecontrol
    - https://www.googleapis.com/auth/service.management.readonly
    - https://www.googleapis.com/auth/trace.append
status: RUNNING
tags:
  fingerprint: 42WmSpB8rSM=
zone: https://www.googleapis.com/compute/v1/projects/wash-247718/zones/us-west1-a
wash . > find . -meta .status RUNNING
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)

natIP: 35.199.171.212
networkTier: PREMIUM
type: ONE_TO_ONE_NAT
fingerprint: VxmZF_2YgX8=
kind: compute#networkInterface
name: nic0
network: https://www.googleapis.com/compute/v1/projects/wash-247718/global/networks/default
networkIP: 10.138.0.3
subnetwork: https://www.googleapis.com/compute/v1/projects/wash-247718/regions/us-west1/subnetworks/default
reservationAffinity:
consumeReservationType: ANY_RESERVATION
scheduling:
automaticRestart: true
onHostMaintenance: MIGRATE
selfLink: https://www.googleapis.com/compute/v1/projects/wash-247718/zones/us-west1-a/instances/michael-test-instance
serviceAccounts:
- email: 668600244557-compute@developer.gserviceaccount.com
  scopes:
  - https://www.googleapis.com/auth/devstorage.read_only
  - https://www.googleapis.com/auth/logging.write
  - https://www.googleapis.com/auth/monitoring.write
  - https://www.googleapis.com/auth/servicecontrol
  - https://www.googleapis.com/auth/service.management.readonly
  - https://www.googleapis.com/auth/trace.append
status: RUNNING
tags:
  fingerprint: 42WmSpB8rSM=
zone: https://www.googleapis.com/compute/v1/projects/wash-247718/zones/us-west1-a
wash . > find . -meta .status RUNNING
./gcp/Wash/compute/michael-test-instance
./gcp/platform-dujour-dev/compute/dujour-dev-3-05281401-9nkj-harness-wp76
wash . >
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)

Recursively descends the directory tree of the specified paths, evaluating an 'expression' composed of 'primaries' and 'operands' for each entry in the tree.

Usage:
  find [paths] [options] [expression]

Flags:
  -depth           Visit the children first before the parent (default false)
  -mdepth depth   Do not print entries at levels less than depth (default 0)
  -maxdepth depth Do not print entries at levels greater than depth (default infinity)
  -daystart        Set the reference time to the start of the current day (default false)
  -fullmeta        Use the entry's full metadata in meta primary predicates (default false)
  -h, --help        Print this usage
  -h, --help <primary> Print a detailed description of the specified primary (e.g. "-help meta")
  -h, --help syntax Print a detailed description of find's expression syntax

Primaries:
  -action action    Returns true if the entry supports action
  -true             Always returns true
  -false            Always returns false
  -k, --kind pattern Returns true if the entry's kind matches pattern
  -m, --meta <expression> Returns true if the entry's metadata satisfies the expression
  -name pattern     Returns true if the entry's cname matches pattern
  -path pattern     Returns true if the entry's normalized path matches pattern
  -size [+|-]n[ckMGTP] Returns true if the entry's size attribute satisfies the given size predicate
  -ctime [+|-]n[smhdw] Returns true if the entry's ctime attribute satisfies the given time predicate
  -mtime [+|-]n[smhdw] Returns true if the entry's mtime attribute satisfies the given time predicate
  -atime [+|-]n[smhdw] Returns true if the entry's atime attribute satisfies the given time predicate
  -crttime [+|-]n[smhdw] Returns true if the entry's crttime attribute satisfies the given time predicate

Operands:
  ( e )            Parentheses operator. Evaluates to true if e evaluates to true
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)

-k, -kind pattern          Returns true if the entry's kind matches pattern
-m, -meta <expression>    Returns true if the entry's metadata satisfies the expression
-name pattern              Returns true if the entry's cname matches pattern
-path pattern              Returns true if the entry's normalized path matches pattern
-size [+|-]n[ckMGTP]       Returns true if the entry's size attribute satisfies the given size predicate
-ctime [+|-]n[smhwd]       Returns true if the entry's ctime attribute satisfies the given time predicate
-mtime [+|-]n[smhwd]       Returns true if the entry's mtime attribute satisfies the given time predicate
-atime [+|-]n[smhwd]       Returns true if the entry's atime attribute satisfies the given time predicate
-crttime [+|-]n[smhwd]    Returns true if the entry's crttime attribute satisfies the given time predicate

Operands:
(e)                         Parentheses operator. Evaluates to true if e evaluates to true
!,-not e                     Logical NOT operator. Evaluates to true if e evaluates to false
e1 -and e2                   Logical AND operator. Evaluates to true if both e1 and e2 are true
e1 -a e2
e1:e2
e1 -or e2                   Logical OR operator. Evaluates to true if either e1 or e2 are true
e1 -o e2

Use "wash find --help <primary>" for more information about a primary. To view
a detailed description of find's expression syntax, use "wash find --help syntax".

NOTE: The default reference time for all time predicates is find's start time.

NOTE: All entry attribute primaries return false if the given entry does not have
the specified attribute. For example, the -mtime primary will always return false
if the entry does not have an mtime attribute.

NOTE: find exits with status 0 if all entries are processed successfully, greater
than 0 if errors occur. This is deliberately a very broad description, but if the
return value is non-zero, you should not rely on the correctness of find.
wash . > find . -meta .status RUNNING -o -meta .State.Name running
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)

Flags:
  -depth           Visit the children first before the parent (default false)
  -mindepth depth Do not print entries at levels less than depth (default 0)
  -maxdepth depth Do not print entries at levels greater than depth (default infinity)
  -daystart        Set the reference time to the start of the current day (default false)
  -fullmeta        Use the entry's full metadata in meta primary predicates (default false)
  -h, --help        Print this usage
  -h, --help <primary> Print a detailed description of the specified primary (e.g. "-help meta")
  -h, --help syntax Print a detailed description of find's expression syntax

Primaries:
  -action action   Returns true if the entry supports action
  -true            Always returns true
  -false           Always returns false
  -k, --kind pattern Returns true if the entry's kind matches pattern
  -m, --meta <expression> Returns true if the entry's metadata satisfies the expression
  -name pattern    Returns true if the entry's cname matches pattern
  -path pattern    Returns true if the entry's normalized path matches pattern
  -size [+|-]n[ckMGTP] Returns true if the entry's size attribute satisfies the given size predicate
  -ctime [+|-]n[smhdw] Returns true if the entry's ctime attribute satisfies the given time predicate
  -mtime [+|-]n[smhdw] Returns true if the entry's mtime attribute satisfies the given time predicate
  -atime [+|-]n[smhdw] Returns true if the entry's atime attribute satisfies the given time predicate
  -crtime [+|-]n[smhdw] Returns true if the entry's crtime attribute satisfies the given time predicate

Operands:
  ( e )           Parentheses operator. Evaluates to true if e evaluates to true
  !, --not e      Logical NOT operator. Evaluates to true if e evaluates to false
  e1 -and e2     Logical AND operator. Evaluates to true if both e1 and e2 are true
  e1 -a e2
  e1 e2
  e1 -or e2     Logical OR operator. Evaluates to true if either e1 or e2 are true
  e1 -o e2
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)

-ctime [+|-]n[smhdw] Returns true if the entry's ctime attribute satisfies the given time predicate
-mtime [+|-]n[smhdw] Returns true if the entry's mtime attribute satisfies the given time predicate
-atime [+|-]n[smhdw] Returns true if the entry's atime attribute satisfies the given time predicate
-crttime [+|-]n[smhdw] Returns true if the entry's crtime attribute satisfies the given time predicate

Operands:
( e ) Parentheses operator. Evaluates to true if e evaluates to true
!e, -not e Logical NOT operator. Evaluates to true if e evaluates to false
e1 -and e2 Logical AND operator. Evaluates to true if both e1 and e2 are true
e1 -a e2
e1 e2
e1 -or e2 Logical OR operator. Evaluates to true if either e1 or e2 are true
e1 -o e2

Use "wash find --help <primary>" for more information about a primary. To view
a detailed description of find's expression syntax, use "wash find --help syntax".

NOTE: The default reference time for all time predicates is find's start time.

NOTE: All entry attribute primaries return false if the given entry does not have
the specified attribute. For example, the -mtime primary will always return false
if the entry does not have an mtime attribute.

NOTE: find exits with status 0 if all entries are processed successfully, greater
than 0 if errors occur. This is deliberately a very broad description, but if the
return value is non-zero, you should not rely on the correctness of find.
wash . > find . -meta .status RUNNING -o -meta .State.Name running
./aws/personal/resources/ec2/instances/i-080ae094d218db0aa
./aws/personal/resources/ec2/instances/i-0f721c2fe68bc082a
./gcp/Wash/compute/michael-test-instance
./gcp/platform-dujour-dev/compute/dujour-dev-3-05281401-9nkj-harness-wp76
wash . > find . -k '/fs'
```





SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help
Default (wash)
./aws/personal/resources/ec2/instances/i-0f721c2fe68bc082a/fs
./docker/containers/k8s_cd4pe-5858774b79-8gbsm_cd4pe_b8843b6c-3a7b-4063-a79b-6b209970a986_0/fs
./docker/containers/k8s_envoy_envoy-trm8k_projectcontour_3fbe5104-999e-4597-ba09-0361f316f1a1_0/fs
./docker/containers/k8s_kotsadm-api_kotsadm-api-678c7b878f-xrlws_default_79535451-64ec-433f-a4d8-76f6522cb0a2_1
./fs
./docker/containers/k8s_kotsadm-minio_kotsadm-minio-0_default_903b0ea3-ff80-48e5-b0ab-0d49f6a91949_0/fs
./docker/containers/k8s_kotsadm-operator_kotsadm-operator-d87c5548b-kwjgt_default_61e3145c-7d89-4e8f-8da5-5ad02
5d3e488_0/fs
./docker/containers/k8s_kotsadm-postgres_kotsadm-postgres-0_default_f977c733-dcde-4a79-8392-1aba67d97dab_0/fs
./docker/containers/k8s_kotsadm_kotsadm-5d9c4c698-sdzxq_default_432bf518-da48-4764-a153-fd28b588747a_0/fs
./docker/containers/k8s_minio_minio-0_cd4pe_3e05a2da-4456-45ea-8920-3a7d0c55f06d_0/fs
./docker/containers/k8s_postgres_postgres-0_cd4pe_be889433-4aeb-4954-b74c-68293938d3d8_0/fs
./gcp/Wash/compute/michael-test-instance/fs
./gcp/platform-dujour-dev/compute/dujour-dev-3-05281401-9nkj-harness-wp76/fs
./kubernetes/docker-desktop/cd4pe/pods/cd4pe-5858774b79-8gbsm/cd4pe/fs
./kubernetes/docker-desktop/cd4pe/pods/minio-0/minio/fs
./kubernetes/docker-desktop/cd4pe/pods/postgres-0/postgres/fs
./kubernetes/docker-desktop/default/pods/kotsadm-5d9c4c698-sdzxq/kotsadm/fs
./kubernetes/docker-desktop/default/pods/kotsadm-api-678c7b878f-xrlws/kotsadm-api/fs
./kubernetes/docker-desktop/default/pods/kotsadm-minio-0/kotsadm-minio/fs
./kubernetes/docker-desktop/default/pods/kotsadm-operator-d87c5548b-kwjgt/kotsadm-operator/fs
./kubernetes/docker-desktop/default/pods/kotsadm-postgres-0/kotsadm-postgres/fs
./kubernetes/docker-desktop/docker/pods/compose-78f95d4f8c-r5w42/compose/fs
./kubernetes/docker-desktop/docker/pods/compose-api-6ffb89dc58-ksvsn/compose/fs
./kubernetes/docker-desktop/kube-system/pods/etcd-docker-desktop/etcd/fs
./kubernetes/docker-desktop/kube-system/pods/kube-apiserver-docker-desktop/kube-apiserver/fs
./kubernetes/docker-desktop/kube-system/pods/kube-controller-manager-docker-desktop/kube-controller-manager/fs
./kubernetes/docker-desktop/kube-system/pods/kube-proxy-5w9n9/kube-proxy/fs
./kubernetes/docker-desktop/kube-system/pods/kube-scheduler-docker-desktop/kube-scheduler/fs
./kubernetes/docker-desktop/kube-system/pods/vpnkit-controller/vpnkit-controller/fs
./kubernetes/docker-desktop/projectcontour/pods/envoy-trm8k/envoy/fs
./kubernetes/docker-desktop/projectcontour/pods/envoy-trm8k/shutdown-manager/fs
```



Classic Tools for the Modern Age: Introducing Wash, the Cloud Native Shell

 onlinexperiences.com/scripts/Server.nxp

SCREEN SHARE

SCREEN SHARE

```
./kubernetes/docker-desktop/cd4pe/pods/cd4pe-5858774b79-8gbsm/cd4pe/fs
./kubernetes/docker-desktop/cd4pe/pods/minio-0/minio/fs
./kubernetes/docker-desktop/cd4pe/pods/postgres-0/postgres/fs
./kubernetes/docker-desktop/default/pods/kotsadm-5d9c4c698-sdzxq/kotsadm/fs
./kubernetes/docker-desktop/default/pods/kotsadm-api-678c7b878f-xrlws/kotsadm-api/fs
./kubernetes/docker-desktop/default/pods/kotsadm-minio-0/kotsadm-minio/fs
./kubernetes/docker-desktop/default/pods/kotsadm-operator-d87c5548b-kwjgt/kotsadm-operator/fs
./kubernetes/docker-desktop/default/pods/kotsadm-postgres-0/kotsadm-postgres/fs
./kubernetes/docker-desktop/docker/pods/compose-78f95d4f8c-r5w42/compose/fs
./kubernetes/docker-desktop/docker/pods/compose-api-6ffb89dc58-ksvsn/compose/fs
./kubernetes/docker-desktop/kube-system/pods/etc-docker-desktop/etc/fs
./kubernetes/docker-desktop/kube-system/pods/kube-apiserver-docker-desktop/kube-apiserver/fs
./kubernetes/docker-desktop/kube-system/pods/kube-controller-manager-docker-desktop/kube-controller-manager/fs
./kubernetes/docker-desktop/kube-system/pods/kube-proxy-5w9n9/kube-proxy/fs
./kubernetes/docker-desktop/kube-system/pods/kube-scheduler-docker-desktop/kube-scheduler/fs
./kubernetes/docker-desktop/kube-system/pods/vpnkit-controller/vpnkit-controller/fs
./kubernetes/docker-desktop/projectcontour/pods/envoy-trm8k/envoy/fs
./kubernetes/docker-desktop/projectcontour/pods/envoy-trm8k/shutdown-manager/fs
^C
wash . > ls
aws/
docker/
gcp/
kubernetes/
wash . > open aws/personal/resources/s3/puppet-wash
wash . > tree gcp/Wash/firestore/
gcp/Wash/firestore/
└── users
    └── bob
        └── data.json
2 directories, 1 file
wash . >
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)

wash . > tree gcp/Wash/firestore/
gcp/Wash/firestore/
└── users
    └── bob
        └── data.json

2 directories, 1 file
wash . > tree gcp/Wash/storage
gcp/Wash/storage
├── artifacts.wash-247718.appspot.com
│   └── containers
│       └── images
│           ├── sha256:41817dae267aee78e765ecbde67ac3360c482cea8e1e420e6fe560dd5c22d001
│           ├── sha256:46201db8d577944e8dda68ffd43ad7ee721973284b92131e56807e588777d6
│           ├── sha256:542388284ed676aec052febcb9363c5f5348630ddecedbe7f08ae811540eb1350
│           └── sha256:89d9c30c1d48bac627e5c6cb0dled1eec28e7dbdfbcc04712e4c79c0f83faf17
├── some-wash-bucket
│   ├── folder
│   │   └── wash.go
│   ├── godoc.sh
│   ├── some-wash-stuff
│   │   ├── Dockerfile
│   │   ├── an\ example\ folder
│   │   │   └── static.sh
│   │   ├── reaper.sh
│   ├── staging.wash-247718.appspot.com
│   ├── wash-247718.appspot.com
│   └── wash-247718_cloudbuild
        └── source
            └── 1574573229.97-0b5654278ec2401b80c0066ff3dc9248.tgz

11 directories, 10 files
wash . > tail -f gcp/Wash/pubsub/my-topic
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)

2 directories, 1 file
wash . > tree gcp/Wash/storage
gcp/Wash/storage
├── artifacts.wash-247718.appspot.com
│   └── containers
│       └── images
│           ├── sha256:41817dae267aee78e765ecbde67ac3360c482cea8e1e420e6fe560dd5c22d001
│           ├── sha256:46201db8d577944e8dda68ffde43ad7ee721973284b932131e56807e588777d6
│           ├── sha256:542388284ed676aec052febcb9363c5f5348630ddecedbe7f08ae811540eb1350
│           └── sha256:89d9c30c1d48bac627e5c6cb0d1ed1eec28e7dbdfbcc04712e4c79c0f83faf17
├── some-wash-bucket
│   ├── folder
│   │   └── wash.go
│   ├── godoc.sh
│   ├── some-wash-stuff
│   ├── Dockerfile
│   ├── an\ example\ folder
│   │   └── static.sh
│   ├── reaper.sh
│   ├── staging.wash-247718.appspot.com
│   ├── wash-247718.appspot.com
│   └── wash-247718_cloudbuild
│       └── source
│           └── 1574573229.97-0b5654278ec2401b80c0066ff3dc9248.tgz
11 directories, 10 files
wash . > tail -f gcp/Wash/pubsub/my-topic&
[1] 51146
wash . > echo hello >> gcp/Wash/pubsub/my-topic
wash . > ===> gcp/Wash/pubsub/my-topic <===
Jun 18 16:25:50.521 | hello

wash . >
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)

└ containers
  └ images
    └ sha256:41817dae267aee78e765ecbde67ac3360c482cea8e1e420e6fe560dd5c22d001
      └ sha256:46201db8d577944e8dda68ffde43ad7ee721973284b932131e56807e588777d6
        └ sha256:542388284ed676aec052feb9363c5f5348639ddecedbe7f08ae811540eb1350
          └ sha256:89d9c30c1d48bac627e5c6cb0d1ed1eec28e7dbdfbcc04712e4c79c0f83faf17

  └ some-wash-bucket
    └ folder
      └ wash.go
    └ godoc.sh
  └ some-wash-stuff
    └ Dockerfile
    └ an\ example\ folder
      └ static.sh
    └ reaper.sh
  └ staging.wash-247718.appspot.com
  └ wash-247718.appspot.com
  └ wash-247718_cloudbuild
    └ source
      └ 1574573229.97-0b5654278ec2401b80c0066ff3dc9248.tgz

11 directories, 10 files
wash . > tail -f gcp/Wash/pubsub/my-topic&
[1] 51146
wash . > echo hello >> gcp/Wash/pubsub/my-topic
wash . > ===> gcp/Wash/pubsub/my-topic <===
Jun 18 16:25:50.521 | hello

wash . >
wash . > fg
[1] + 51146 running      WASH_EMBEDDED=1 wash tail -f gcp/Wash/pubsub/my-topic
^C
wash . >
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)

11 directories, 10 files
wash . > tail -f gcp/Wash/pubsub/my-topic&
[1] 51146
wash . > echo hello >> gcp/Wash/pubsub/my-topic
wash . > ===> gcp/Wash/pubsub/my-topic <===
Jun 18 16:25:50.521 | hello

wash . >
wash . > fg
[1] + 51146 running      WASH_EMBEDDED=1 wash tail -f gcp/Wash/pubsub/my-topic
^C
wash . > ls
aws/
docker/
gcp/
kubernetes/
wash . > cd docker
wash docker > cd containers
wash docker/containers > ls
k8s_cd4pe_cd4pe-5858774b79-8gbsm_cd4pe_b8843b6c-3a7b-4063-a79b-6b209970a986_0/
k8s_contour_contour-77786b64df-kjpg8_projectcontour_3286d41f-c087-4dfa-a6c5-3b68909ffdc4_0/
k8s_contour_contour-77786b64df-xw2mq_projectcontour_cb0d485a-9e06-4f4b-a6b8-056cc65776ec_0/
k8s_envoy_envoy-trm8k_projectcontour_3fbe5104-999e-4597-ba09-0361f316f1a1_0/
k8s_kotsadm-api_kotsadm-api-678c7b878f-xrlws_default_79535451-64ec-433f-a4d8-76f6522cb0a2_1/
k8s_kotsadm-minio_kotsadm-minio-0_default_903b0ea3-ff80-48e5-b0ab-0d49f6a91949_0/
k8s_kotsadm-operator_kotsadm-operator-d87c5548b-kwjgt_default_61e3145c-7d89-4e8f-8da5-5ad025d3e488_0/
k8s_kotsadm-postgres_kotsadm-postgres-0_default_f977c733-dcde-4a79-8392-1aba67d97dab_0/
k8s_kotsadm_kotsadm-5d9c4c698-sdzxq_default_432bf518-da48-4764-a153-fd28b588747a_0/
k8s_minio_minio-0_cd4pe_3e05a2da-4456-45ea-8920-3a7d0c55f06d_0/
k8s_postgres_postgres-0_cd4pe_be889433-4aeb-4954-b74c-68293938d3d8_0/
k8s_shutdown-manager_envoy-trm8k_projectcontour_3fbe5104-999e-4597-ba09-0361f316f1a1_0/
wash docker/containers >
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)
7cf7e4c5d5c51cae7da32cae1f5a1be6c91cdæ3323961b875c2983239dd4777/
94ee81cccd0b7a6b63e7cf6ec6d61c645db91262af8c4b03ca9763a3bd1910546/
9d9f3dba4dacb634ba21622836b8e71780da308bb0c7ef1bdc4d84d6f2b2857b/
a755d61d48284eb2b2213c8c96f0f1056ea30e3cbe0faab4b31a44dfe8b9c61f/
a8e7b44777da85a370d20d0f15a158c5c3ee713fc0014d3aecd548158c17b46/
ade310e6299427436a3a913b9c6a449ed131c20f412864bb467c6da5b847fd26/
aef850886666cf82f830dd17bef6f3e3472e0a943ff95726fc8e28ed3d2935432/
b68635f7828559bafeed39f86ccc8c12380aae1f3288ac9ad1f8bfd5ab4d9d57/
c482d25d57c19180f3246c9a5fcbc8fc6c11f154639c902ecce2133ee2fad74d/
c5b941be27d3060262a187123a6e7b35b5503697cd92de6d0bdbcd1d000c4f8f/
c69dbe9d2d709ac3ed8d211508cd4e617c10665833855c7245d0298295003e94/
cd4pe/
cd4pe-postgres/
d5b9248cb8886c121ffa6d97978092dbdcc27796e2a76edefe6aef00c61234da/
e34eebe7952eb5271d2af29b31fba504a7e9438eea6694d4d2549028f591837b/
e69333832ec7776989345589f5682cde58a51ed02efd62b14e3cab12d083c66e/
eb681cdle0dc952407abd9f8cdbfdal5b4a97f3f60fd2dc62f03336374e00b/
ebac824f4a109488c740013be14c4bae40870c3194075ccc81c753e96ecb5706/
keycloak/
pipelinesinfra_cd4pe-mysql/
pipelinesinfra_data/
pupperware_cd4pe-data/
pupperware_cd4pe-postgres/
pupperware_puppetdb/
pupperware_puppetdb-postgres/
pupperware_puppetserver-code/
pupperware_puppetserver-config/
pupperware_puppetserver-data/
swarm_redis/
wash docker/volumes > ls swarm_redis
appendonly.aof
foo
wash docker/volumes > delete swarm_redis/
```



SCREEN SHARE

```
SELECT
$1
0
*3
$6
INCRBY
$4
hits
$1
1
*2
$6
SELECT
$1
0
*3
$6
INCRBY
$4
hits
$1
1
*3
$6
INCRBY
$4
hits
$1
1
wash docker/volumes > delete swarm_redis/foo
remove swarm_redis/foo? y
swarm_redis/foo has been deleted
wash docker/volumes >
```



SCREEN SHARE

```
$6
INCRBY
$4
hits
$1
1
*2
$6
SELECT
$1
0
*3
$6
INCRBY
$4
hits
$1
1
*3
$6
INCRBY
$4
hits
$1
1
wash docker/volumes > delete swarm_redis/foo
remove swarm_redis/foo? y
swarm_redis/foo has been deleted
wash docker/volumes > ls swarm_redis
appendonly.aof
wash docker/volumes > cd ..
wash docker > wps containers/*
-- no matches found --
```





SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)

.00  /kotsadm-operator
cos/s/k8s_kotsadm-postgres_kotsadm-postgres-0_default_f977c733-dcde-4a79-8392-1aba67d97dab_0 1 00:00
.00  postgres
cos/s/k8s_kotsadm-postgres_kotsadm-postgres-0_default_f977c733-dcde-4a79-8392-1aba67d97dab_0 56 00:00
.00  postgres: checkpointer process
cos/s/k8s_kotsadm-postgres_kotsadm-postgres-0_default_f977c733-dcde-4a79-8392-1aba67d97dab_0 57 00:00
.00  postgres: writer process
cos/s/k8s_kotsadm-postgres_kotsadm-postgres-0_default_f977c733-dcde-4a79-8392-1aba67d97dab_0 58 00:00
.00  postgres: wal writer process
cos/s/k8s_kotsadm-postgres_kotsadm-postgres-0_default_f977c733-dcde-4a79-8392-1aba67d97dab_0 59 00:00
.00  postgres: autovacuum launcher process
cos/s/k8s_kotsadm-postgres_kotsadm-postgres-0_default_f977c733-dcde-4a79-8392-1aba67d97dab_0 60 00:00
.00  postgres: stats collector process
cos/s/k8s_kotsadm-postgres_kotsadm-postgres-0_default_f977c733-dcde-4a79-8392-1aba67d97dab_0 61 00:00
.00  postgres: bgworker: logical replication launcher
cos/s/k8s_kotsadm-postgres_kotsadm-postgres-0_default_f977c733-dcde-4a79-8392-1aba67d97dab_0 137 00:00
.00  postgres: kotsadm kotsadm 10.1.4.36(59214) idle
cos/s/k8s_kotsadm-postgres_kotsadm-postgres-0_default_f977c733-dcde-4a79-8392-1aba67d97dab_0 300 00:00
.00  postgres: kotsadm kotsadm 10.1.4.36(59486) idle
cos/s/k8s_kotsadm-postgres_kotsadm-postgres-0_default_f977c733-dcde-4a79-8392-1aba67d97dab_0 318 00:00
.00  postgres: kotsadm kotsadm 10.1.4.37(53086) idle
cos/s/k8s_kotsadm-postgres_kotsadm-postgres-0_default_f977c733-dcde-4a79-8392-1aba67d97dab_0 48105 00:00
.00  postgres: kotsadm kotsadm 10.1.4.37(44326) idle
cos/s/k8s_kotsadm-postgres_kotsadm-postgres-0_default_f977c733-dcde-4a79-8392-1aba67d97dab_0 51358 00:00
.00  postgres: kotsadm kotsadm 10.1.4.37(49498) idle
cos/s/k8s_kotsadm-postgres_kotsadm-postgres-0_default_f977c733-dcde-4a79-8392-1aba67d97dab_0 76245 00:00
.00  postgres: kotsadm kotsadm 10.1.4.37(60422) idle
cos/s/k8s_kotsadm_kotsadm-5d9c4c698-sdzxq_default_432bf518-da48-4764-a153-fd28b588747a_0 1 00:00
.00  /kotsadm api
cos/s/k8s_minio_minio-0_cd4pe_3e05a2da-4456-45ea-8920-3a7d0c55f06d_0 1 00:00
.00  minio server /data
cos/s/k8s_postgres_postgres-0_cd4pe_be889433-4aeb-4954-b74c-68293938d3d8_0 1 00:00
.00  postgres
```





SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)

.00  postgres: stats collector process
co*s/k8s_postgres_postgres-0_cd4pe_be889433-4aeb-4954-b74c-68293938d3d8_0 69791 00:00
.00  postgres: cd4pe cd4pe 10.1.4.41(50050) idle
co*s/k8s_postgres_postgres-0_cd4pe_be889433-4aeb-4954-b74c-68293938d3d8_0 69932 00:00
.00  postgres: cd4pe cd4pe 10.1.4.41(50274) idle
co*s/k8s_postgres_postgres-0_cd4pe_be889433-4aeb-4954-b74c-68293938d3d8_0 69933 00:00
.00  postgres: cd4pe cd4pe 10.1.4.41(50276) idle
co*s/k8s_postgres_postgres-0_cd4pe_be889433-4aeb-4954-b74c-68293938d3d8_0 69972 00:00
.00  postgres: cd4pe cd4pe 10.1.4.41(50334) idle
co*s/k8s_postgres_postgres-0_cd4pe_be889433-4aeb-4954-b74c-68293938d3d8_0 70033 00:00
.00  postgres: cd4pe cd4pe 10.1.4.41(50432) idle
co*s/k8s_postgres_postgres-0_cd4pe_be889433-4aeb-4954-b74c-68293938d3d8_0 70041 00:00
.00  postgres: cd4pe cd4pe 10.1.4.41(50450) idle
co*s/k8s_postgres_postgres-0_cd4pe_be889433-4aeb-4954-b74c-68293938d3d8_0 70143 00:00
.00  postgres: cd4pe cd4pe 10.1.4.41(50608) idle
co*s/k8s_postgres_postgres-0_cd4pe_be889433-4aeb-4954-b74c-68293938d3d8_0 70195 00:00
.00  postgres: cd4pe cd4pe 10.1.4.41(50694) idle
co*s/k8s_postgres_postgres-0_cd4pe_be889433-4aeb-4954-b74c-68293938d3d8_0 70231 00:00
.00  postgres: cd4pe cd4pe 10.1.4.41(50750) idle
co*s/k8s_postgres_postgres-0_cd4pe_be889433-4aeb-4954-b74c-68293938d3d8_0 70382 00:00
.00  postgres: cd4pe cd4pe 10.1.4.41(50996) idle
wash docker > cd ..
wash . > ls
aws/
docker/
gcp/
kubernetes/
wash . > cd kubernetes/docker-desktop/cd4pe
wash kubernetes/docker-desktop/cd4pe > pods
wash kubernetes/docker-desktop/cd4pe/pods > tail -f */*/log
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (wash)

Error: Bucket not found: .minio.sys/buckets
  1: github.com/minio/minio@/cmd/data-usage.go:69:cmd.runDataUsageInfo()

====> cd4pe-5858774b79-8gbsm/cd4pe/log <===
2020-06-17 22:26:10.380 UTC:[DEBUG]:[qtp1491445403-106]:com.puppet.cdpe.objectStore.impl.s3.S3ObjectStore:User supplied environment override for s3 path-style access: true
2020-06-17 22:26:20.356 UTC:[DEBUG]:[qtp1491445403-106]:com.puppet.cdpe.objectStore.impl.s3.S3ObjectStore:User supplied environment override for s3 path-style access: true
2020-06-17 22:26:23.598 UTC:[DEBUG]:[qtp1491445403-91]:com.puppet.cdpe.objectStore.impl.s3.S3ObjectStore:User supplied environment override for s3 path-style access: true
2020-06-17 22:26:24.534 UTC:[DEBUG]:[qtp1491445403-91]:com.puppet.cdpe.objectStore.impl.s3.S3ObjectStore:User supplied environment override for s3 path-style access: true
2020-06-17 22:26:29.627 UTC:[INFO]:[AgentService]:com.distelli.agent.GuiceAgentService:Starting AgentService listener on dump://cd4pe:7000/
2020-06-18 05:30:24.682 UTC:[WARN]:[HikariPool-1 housekeeper]:com.zaxxer.hikari.pool.HikariPool:HikariPool-1 - Thread starvation or clock leap detected (housekeeper delta=1m45s157ms808μs700ns).
2020-06-18 05:30:24.551 UTC:[ERROR]:[BackgroundTaskRunner-1-1896]:com.distelli.monitor.impl.MonitorInfoImpl:Forcing heart beat failure due to max time between heartbeat
2020-06-18 05:30:24.953 UTC:[WARN]:[BackgroundTaskRunner-1-1896]:com.distelli.monitor.impl.MonitorImpl:Adjusting waitTime =-27475ms to 200ms when running shutdown(1vb1e86asi3oe0q42lwr22ef88)

====> postgres-0/postgres/log <===
LOG:  database system is shut down
done
server stopped

PostgreSQL init process complete; ready for start up.

LOG:  database system was shut down at 2020-06-17 22:21:38 UTC
LOG:  MultiXact member wraparound protections are now enabled
LOG:  database system is ready to accept connections
LOG:  autovacuum launcher started
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)

wash kubernetes > cd ..
wash . > ls
aws/
docker/
gcp/
kubernetes/
wash . > help
Type 'exit' to exit the shell at any time.

Usage:
  [command]

Available Commands:
  delete      Deletes the entries at the specified paths
  docs        Displays the entry's documentation
  find        Prints out all entries that satisfy the given expression
  help        Help about any command
  ls          Lists the children of the specified paths, or current directory if not specified
  meta        Prints the metadata of the given entries
  signal      Sends the specified signal to the entries at the specified paths
  stree       Displays the entry's stree (schema-tree)
  tail        Displays new output of files or resources with the stream action
  wclear      Clears the cache at the specified paths, or current directory if not specified
  wexec       Executes the given command on the indicated target
  whistory    Prints the wash command history, or journal of a particular item
  winfo       Wash's version of 'stat'. Prints the entries' info at the specified paths
  wps         Lists the processes running on the indicated compute instances

Flags:
  -h, --help  help for this command

Use " [command] --help" for more information about a command.
wash . >
```





SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)
wash . > whistory 39
Jun 18 09:27:40.838 API: GET /fs/stream?path=%2FUsers%2Fmichaelsmith%2FLibrary%2FCaches%2Fwash%2Fmnt204257757%2Fkubernetes%2Fdocker-desktop%2Fc4pe%2Fpods%2Fc4pe-5858774b79-8gbsm%2Fc4pe%2Flog
Jun 18 09:27:40.850 Listing namespaces: [      ]
Jun 18 09:27:40.874 API: Streaming /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/cd4pe-5858774b79-8gbsm/cd4pe/log
Jun 18 09:27:40.874 API: GET /fs/stream?path=%2FUsers%2Fmichaelsmith%2FLibrary%2FCaches%2Fwash%2Fmnt204257757%2Fkubernetes%2Fdocker-desktop%2Fc4pe%2Fpods%2Fminio-0%2Fminio%2Flog
Jun 18 09:27:40.902 API: Streaming /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/minio-0/minio/log
Jun 18 09:27:40.904 API: GET /fs/stream?path=%2FUsers%2Fmichaelsmith%2FLibrary%2FCaches%2Fwash%2Fmnt204257757%2Fkubernetes%2Fdocker-desktop%2Fc4pe%2Fpods%2Fpostgres-0%2Fpostgres%2Flog
Jun 18 09:27:40.933 API: Streaming /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/postgres-0/postgres/log
Jun 18 09:27:53.651 API: Stream /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/postgres-0/postgres/log closed by completed context: <nil>
Jun 18 09:27:53.651 API: Stream /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/minio-0/minio/log closed by completed context: <nil>
Jun 18 09:27:53.651 API: Streaming /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/minio-0/minio/log errored: http2: response body closed
Jun 18 09:27:53.651 API: Streaming /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/postgres-0/postgres/log errored: http2: response body closed
Jun 18 09:27:53.651 API: Stream /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/cd4pe-5858774b79-8gbsm/cd4pe/log errored: http2: response body closed
Jun 18 09:27:53.651 API: GET /fs/stream?path=%2FUsers%2Fmichaelsmith%2FLibrary%2FCaches%2Fwash%2Fmnt204257757%2Fkubernetes%2Fdocker-desktop%2Fc4pe%2Fpods%2Fpostgres-0%2Fpostgres%2Flog complete
Jun 18 09:27:53.651 API: Stream /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/cd4pe-5858774b79-8gbsm/cd4pe/log closed by completed context: <nil>
Jun 18 09:27:53.651 API: GET /fs/stream?path=%2FUsers%2Fmichaelsmith%2FLibrary%2FCaches%2Fwash%2Fmnt204257757%2Fkubernetes%2Fdocker-desktop%2Fc4pe%2Fpods%2Fc4pe-5858774b79-8gbsm%2Fc4pe%2Flog complete
Jun 18 09:27:53.651 API: GET /fs/stream?path=%2FUsers%2Fmichaelsmith%2FLibrary%2FCaches%2Fwash%2Fmnt204257757%2Fkubernetes%2Fdocker-desktop%2Fc4pe%2Fpods%2Fminio-0%2Fminio%2Flog complete
wash . >
```





VIDEO



ABSTRACT

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.

SLIDES

```
wash > cd aws/wash/resources
wash > ls
ec2/
s3/
wash > cd ec2/instances
wash > ls
i-04621c13583930e6c/
wash > find . -k '*instance' -m '.state.name' running -m '.tags[?].key' owner
./i-04621c13583930e6c
wash > wexec i-04621c13583930e6c uname
Linux
wash > cd i-04621c13583930e6c
```

AWS





VIDEO



ABSTRACT

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.

SLIDES

```
wash > cd spotify
wash > ls
playlists/
wash > cd playlists
wash > ls
classical/
gym/
wash > cd gym
wash > ls
Down with the Sickness
Remember the Name (feat. Styles of Beyond)
wash > find . -m '.artist' Disturbed
./Down with the Sickness
wash > cat Down\ with\ the\ Sickness
```

Spotify





VIDEO



SLIDES

Welcome to Wash!

Wash includes several built-in commands: wexec, find, list, meta, tail.
See commands run with wash via 'whistory', and logs with 'whistory <id>'.

Try 'help'

wash . > help

Wash sets the 'W' environment variable to its starting point. Use \$W to return to it.

Usage:

[command]

Available Commands:

delete	Deletes the entries at the specified paths
docs	Displays the entry's documentation. This is currently its description and any supported signals/signal groups.
find	Prints out all entries that satisfy the given expression
help	Help about any command
ls	Lists the resources at the indicated path
meta	Prints the entry's metadata
signal	Sends the specified signal to the entry at the specified path
stree	Displays the entry's stree (schema-tree)
tail	Displays new output of files or resources with the stream action
version	Print wash version
wclear	Clears the cache at <path>, or current directory if not specified
wexec	Executes the given command on the indicated target
whistory	Prints the wash command history, or journal of a particular item
winfo	Prints the entry's info at the specified path
wps	Lists the processes running on the indicated compute instances

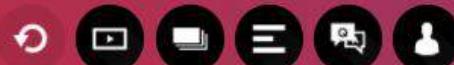
Flags:

-h, --help help for this command

Use " [command] --help" for more information about a command.

pup.pt/wash

14





VIDEO

**ABSTRACT**

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.

SLIDES

```

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 sab4k vieux
chealer@vinci:~$ why is there color when calling ls without arguments?
chealer@vinci:~$ which ls
/bin/ls
chealer@vinci:~$ $(!!)
$(which ls)
Cloutier Ido Musique logs skolo sources
Desktop Mes images boston ncix.png sab4k vieux
chealer@vinci:~$ type ls #`ls` doesn't just run /bin/ls
ls is aliased to ls --color=auto
chealer@vinci:~$ echo $PS1
$(debian_chroot:+($debian_chroot)}\w@\h:\w\$
chealer@vinci:~$ sh
sh-3.1$ echo $PS1
\$-\V\$
sh-3.1$ echo $BASH_VERSION
3.1.17(1)-release
sh-3.1$ ls
Cloutier Ido Musique logs skolo sources
Desktop Mes images boston ncix.png sab4k 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 killerror
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
- Scriptable
- User-focused design
- Hierarchical navigation
- Locational context



VIDEO



ABSTRACT

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.

SLIDES

```
$ echo $VAULT_ADDR $VAULT_TOKEN
http://localhost:8889 root
$ ./vsh
http://localhost:8889 /> ls
secretkv1/ secretkv2/
http://localhost:8889 /> cd secretkv1/
http://localhost:8889 /secretkv1/> ls
department8/
http://localhost:8889 /secretkv1/
```

```
cd   cd <path>
cp   cp <from> <to> | -r is implied
rm   rm <path> | -r is implied
mv   mv <from> <to>
cat  cat <path>
ls   ls <path>
```

github.com/fishi0x01/vsh



**VIDEO****ABSTRACT**

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.

SLIDES

```
$ echo $VAULT_ADDR $VAULT_TOKEN  
http://localhost:8889 root  
$ ./vsh  
http://localhost:8889 /> ls  
secretkv1/ secretkv2/  
http://localhost:8889 /> cd secretkv1/  
http://localhost:8889 /secretkv1/> ls  
departmentB/  
http://localhost:8889 /secretkv1/> cat departmentB/s
```

```
departmentB/secretA  
departmentB/secretB  
departmentB/subsecret/
```

github.com/fishi0x01/vsh



SCREEN SHARE

```
$ echo $VAULT_ADDR $VAULT_TOKEN
http://localhost:8889 root
$ ./vsh
http://localhost:8889 > ls
secretkv1/ secretkv2/
http://localhost:8889 > cd secretkv1/
http://localhost:8889 /secretkv1> ls
departmentB/
http://localhost:8889 /secretkv1/>
```

vsh is an interactive [HashiCorp Vault](#) shell which treats paths and keys like directories and files. Key features are:

- recursive operations on paths with `cp`, `mv` OR `rm`
- term search with `grep`
- transparency towards differences between KV1 and KV2, i.e., you can freely move/copy secrets between both
- non-interactive mode for automation (`vsh -c "<cmd>"`)
- merging keys with different strategies through `append`

Supported commands



SCREEN SHARE

The screenshot shows a web browser window with a dark theme. At the top, there are three tabs: 'Open Source Summit + Emby', 'GitHub - fishi0x01/vsh: vsh - H...', and a '+' button. The main content area has a header 'Status' with a green badge for 'release v0.6.1', a green badge for 'CI passing', a blue badge for 'go report', and a green badge for 'maintainability A'. Below this is a section titled 'vsh' containing a terminal session:

```
$ echo $VAULT_ADDR $VAULT_TOKEN
http://localhost:8889 root
$ ./vsh
http://localhost:8889 /> ls
secretkv1/ secretkv2/
http://localhost:8889 /> cd secretkv1/
http://localhost:8889 /secretkv1/> ls
departmentB/
http://localhost:8889 /secretkv1/> cat departmentB/subsecret/secretC
value = secretC
http://localhost:8889 /secretkv1/> mv departmentB/ /secretkv2/departmentB
Moved /secretkv1/departmentB/secretA to /secretkv2/departmentB/secretA
Moved /secretkv1/departmentB/secretB to /secretkv2/departmentB/secretB
Moved /secretkv1/departmentB/subsecret//secretC to /secretkv2/departmentB/subsecret//secretC
http://localhost:8889 /secretkv1/> cat /secretkv2/departmentB/
```

The last few lines of the terminal output are highlighted with a green background.

`vsh` is an interactive [HashiCorp Vault](#) shell which treats paths and keys like directories and files. Key features are:

- recursive operations on paths with `cp`, `mv` or `rm`
- term search with `grep`
- transparency towards differences between KV1 and KV2, i.e., you can freely move/copy secrets between both
- non-interactive mode for automation (`vsh -c "<cmd>"`)
- merging keys with different strategies through `append`

Supported commands





VIDEO



SLIDES

Cloud-native System Calls

ABSTRACT

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.

17





VIDEO



SLIDES

Wash Primitives

List/Metadata

Read/Stream

Write

Exec

Delete

Signal

...

18

ABSTRACT

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.





VIDEO

**ABSTRACT**

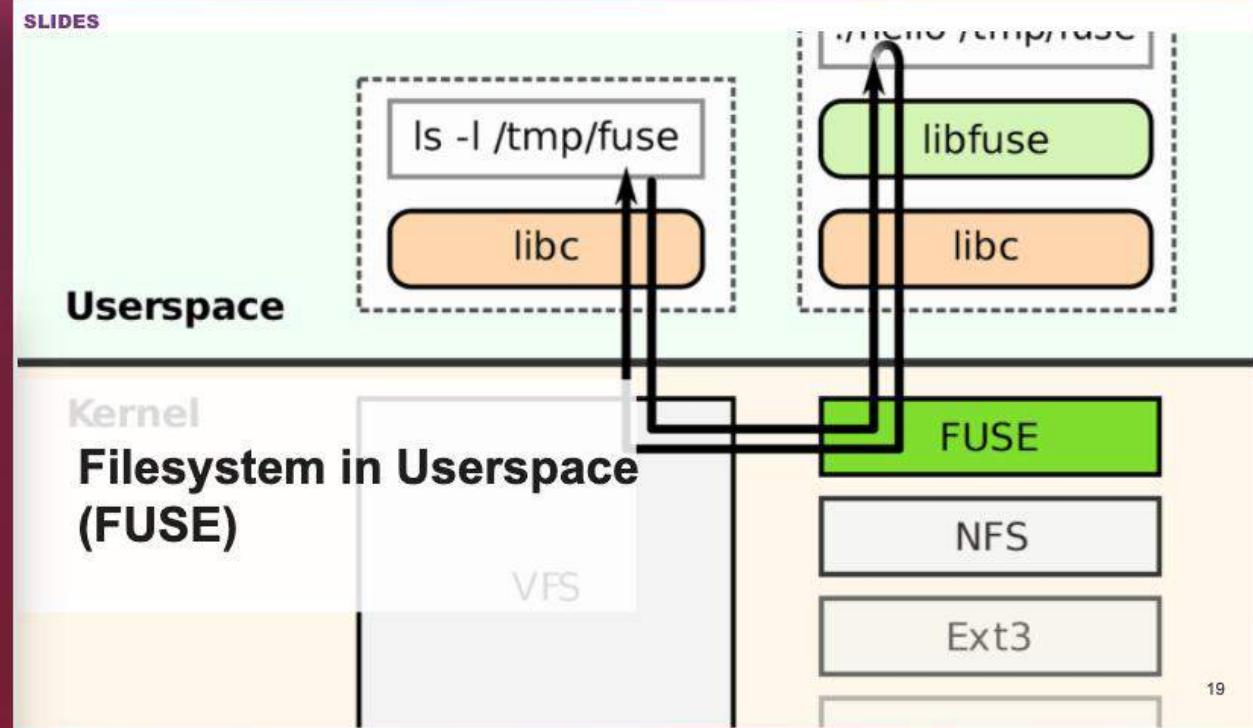
Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.

SLIDES

Userspace**Kernel****Filesystem in Userspace
(FUSE)**



VIDEO



SLIDES

wash ls

wash exec

wash ps

libfuse

bash, zsh

socket

wash

aws plugin

bolt plugin

k8s plugin

ABSTRACT

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.

20





VIDEO



SLIDES

Plugins

```
<plugin_script> <method> <path> <state> <args...>
```

where

- **<method>** is the Wash method that's being invoked. This includes Wash actions like `list` and `exec`, and also non-Wash actions like `schema` and `metadata`.
- **<path>** is the entry's filesystem path rooted at Wash's mountpoint. For example, `/my_plugin` would be the `my_plugin`'s plugin root. `/my_plugin/foo` would be the `foo` child of the `my_plugin` entry.
- **<state>** consists of the minimum amount of information required to reconstruct the entry inside the plugin. It can be *any* string. For example, `{"klass": "Root"}` could be a JSON object representing the plugin root in a higher-level programming language like Python or Ruby.
- **<args...>** are **<method>**'s arguments. For example if **<method>** is `exec`, then the exec'd command would be included in **<args...>**.

puppetlabs.github.io/wash/docs/external-plugins

21

ABSTRACT

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.



SCREEN SHARE

```
1 external-plugins:
2 - script: '/Users/michaelsmith/puppetlabs/washreads/goodreads'
3
4 aws:
5   profiles: [personal]
6 gcp:
7   projects: [Wash, platform-dujour-dev, 'Cpp Projects']
8
9 plugins:
10 - gcp
11 - kubernetes
12 - aws
13 - docker
```



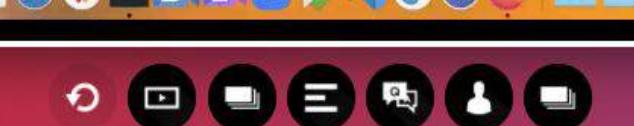


SCREEN SHARE

iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (wash)

```
Jun 18 09:27:40.902 API: Streaming /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/minio-0/minio/log
Jun 18 09:27:40.904 API: GET /fs/stream?path=%2FUsers%2Fmichaelsmith%2FLibrary%2FCaches%2Fwash%2Fmnt204257757%2Fkubernetes%2Fdocker-desktop%2Fcd4pe%2Fpods%2Fpostgres-0%2Fpostgres%2Flog
Jun 18 09:27:40.933 API: Streaming /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/postgres-0/postgres/log
Jun 18 09:27:53.651 API: Stream /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/postgres-0/postgres/log closed by completed context: <nil>
Jun 18 09:27:53.651 API: Stream /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/minio-0/minio/log closed by completed context: <nil>
Jun 18 09:27:53.651 API: Streaming /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/minio-0/minio/log errored: http2: response body closed
Jun 18 09:27:53.651 API: Streaming /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/postgres-0/postgres/log errored: http2: response body closed
Jun 18 09:27:53.651 API: Streaming /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/cd4pe-5858774b79-8gbsm/cd4pe/log errored: http2: response body closed
Jun 18 09:27:53.651 API: GET /fs/stream?path=%2FUsers%2Fmichaelsmith%2FLibrary%2FCaches%2Fwash%2Fmnt204257757%2Fkubernetes%2Fdocker-desktop%2Fcd4pe%2Fpods%2Fpostgres-0%2Fpostgres%2Flog complete
Jun 18 09:27:53.651 API: Stream /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/cd4pe-5858774b79-8gbsm/cd4pe/log closed by completed context: <nil>
Jun 18 09:27:53.651 API: GET /fs/stream?path=%2FUsers%2Fmichaelsmith%2FLibrary%2FCaches%2Fwash%2Fmnt204257757%2Fkubernetes%2Fdocker-desktop%2Fcd4pe%2Fpods%2Fcd4pe-5858774b79-8gbsm%2Fcd4pe%2Flog complete
Jun 18 09:27:53.651 API: GET /fs/stream?path=%2FUsers%2Fmichaelsmith%2FLibrary%2FCaches%2Fwash%2Fmnt204257757%2Fkubernetes%2Fdocker-desktop%2Fcd4pe%2Fpods%2Fminio-0%2Fminio%2Flog complete
wash . > exit
Goodbye!
```

Applications/ Documents/ Dropbox/ Library/ Music/ Public/ keycloak/
Desktop/ Downloads/ Google Drive/ Movies/ Pictures/ VMs/ puppetlabs/
> wash
Welcome to Wash! Try 'docs .'





SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)

Jun 18 09:27:53.651 API: Streaming /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/cd4pe-5858774b79-8gbsm/cd4pe/log errored: http2: response body closed
Jun 18 09:27:53.651 API: GET /fs/stream?path=%2FUsers%2Fmichaelsmith%2FLibrary%2FCaches%2Fwash%2Fmnt204257757%2Fkubernetes%2Fdocker-desktop%2Fc4pe%2Fpods%2Fpostgres-0%2Fpostgres%2Flog complete
Jun 18 09:27:53.651 API: Stream /Users/michaelsmith/Library/Caches/wash/mnt204257757/kubernetes/docker-desktop/cd4pe/pods/cd4pe-5858774b79-8gbsm/cd4pe/log closed by completed context: <nil>
Jun 18 09:27:53.651 API: GET /fs/stream?path=%2FUsers%2Fmichaelsmith%2FLibrary%2FCaches%2Fwash%2Fmnt204257757%2Fkubernetes%2Fdocker-desktop%2Fc4pe%2Fpods%2Fc4pe-5858774b79-8gbsm%2Fc4pe%2Flog complete
Jun 18 09:27:53.651 API: GET /fs/stream?path=%2FUsers%2Fmichaelsmith%2FLibrary%2FCaches%2Fwash%2Fmnt204257757%2Fkubernetes%2Fdocker-desktop%2Fc4pe%2Fpods%2Fminio-0%2Fminio%2Flog complete
wash . > exit
Goodbye!
vi ~/.puppetlabs/wash/wash.yaml
ls
Applications/ Documents/ Dropbox/ Library/ Music/ Public/
Desktop/ Downloads/ Google Drive/ Movies/ Pictures/ VMs/ keycloak/
~ wash
Welcome to Wash! Try 'docs .'
wash . > ls
aws/
docker/
gcp/
goodreads/
kubernetes/
wash . > cd goodreads
ls
wash goodreads > ls
currently-reading/
fantasy/
read/
science-fiction/
to-read/
wash goodreads >
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)

Thud! (Discworld, #34; City Watch #7)
Thuvia, Maid of Mars (Barsoom, #4)
Time of the Twins (Dragonlance: Legends, #1)
Tongues of Serpents (Temeraire, #6)
Towers of Midnight (The Wheel of Time, #13)
Triplanetary (Lensman, #1)
Turn Coat (The Dresden Files, #11)
Unseen Academicals (Discworld, #37; Rincewind #8)
Upland Outlaws (A Handful of Men, #2)
Use of Weapons (Culture #3)
Veiled (Alex Verus, #6)
Victory of Eagles (Temeraire, #5)
Virtual Light (Bridge, #1)
War of Honor (Honor Harrington, #10)
War of the Twins (Dragonlance: Legends, #2)
Wards of Faerie (The Dark Legacy of Shannara, #1)
When True Night Falls (The Coldfire Trilogy, #2)
White Night (The Dresden Files, #9)
Winter's Heart (The Wheel of Time, #9)
Winterfair Gifts (Vorkosigan Saga)
Wintersmith (Discworld, #35; Tiffany Aching, #3)
Witch Wraith (The Dark Legacy of Shannara, #3)
Witches Abroad (Discworld, #12; Witches #3)
Witches' Brew (Magic Kingdom of Landover, #5)
Wizard at Large (Magic Kingdom of Landover #3)
Wizard's First Rule (Sword of Truth, #1)
Wyrd Sisters (Discworld, #6; Witches #2)
Wyrd Sisters: The Play
Xenocide (Ender's Saga, #3)
Young Miles (Vorkosigan Omnibus, #2)
Zero History (Blue Ant, #3)
Zoe's Tale (Old Man's War, #4)
wash goodreads > meta read/Zoe\ 's\ Tale\ \Old\ Man\ 's\ War,\ \#4\)
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)

4763
John Scalzi

https://images.gr-assets.com/authors/1562613145p5/4763.jpg

https://images.gr-assets.com/authors/1562613145p2/4763.jpg
https://www.goodreads.com/author/show/4763.John_Scalzi
4.03
719951
57916

average_rating: "3.74"
description: 'How do you tell your part in the biggest tale in history?<br /><br />I ask because it''s what I have to do. I''m Zoe Boutin Perry: A colonist stranded on a deadly pioneer world. Holy icon to a race of aliens. A player (and a pawn) in a interstellar chess match to save humanity, or to see it fall. Witness to history. Friend. Daughter. Human. Seventeen years old.<br /><br />Everyone on Earth knows the tale I am part of. But you don''t know my tale: How I did what I did - how I did what I had to do - not just to stay alive but to keep you alive, too. All of you. I''m going to tell it to you now, the only way I know how: not straight but true, the whole thing, to try make you feel what I felt: the joy and terror and uncertainty, panic and wonder, despair and hope. Everything that happened, bringing us to Earth, and Earth out of its captivity. All through my eyes.<br /><br />It''s a story you know. But you don''t know it all.'
edition_information: Reprint Edition
id: "8156551"
image_url: https://s.gr-assets.com/assets/nophoto/book/111x148-bcc042a9c91a29cid680899eff700a03.png
link: https://www.goodreads.com/book/show/8156551-zoe-s-tale
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)
average_rating: "3.74"
description: 'How do you tell your part in the biggest tale in history?<br /><br />I ask because it''s what I have to do. I''m Zoe Boutin Perry: A colonist stranded on a deadly pioneer world. Holy icon to a race of aliens. A player (and a pawn) in a interstellar chess match to save humanity, or to see it fall. Witness to history. Friend. Daughter. Human. Seventeen years old.<br /><br />Everyone on Earth knows the tale I am part of. But you don't know my tale: How I did what I did – how I did what I had to do – not just to stay alive but to keep you alive, too. All of you. I'm going to tell it to you now, the only way I know how: not straight but true, the whole thing, to try make you feel what I felt: the joy and terror and uncertainty, panic and wonder, despair and hope. Everything that happened, bringing us to Earth, and Earth out of its captivity. All through my eyes.<br /><br />It's a story you know. But you don't know it all.'
edition_information: Reprint Edition
id: "8156551"
image_url: https://s.gr-assets.com/assets/nophoto/book/111x148-bcc042a9c91a29c1d680899eff700a03.png
link: https://www.goodreads.com/book/show/8156551-zoe-s-tale
num_pages: "420"
publication_day: "1"
publication_month: "4"
publication_year: "2010"
published: "2010"
publisher: Tor Books
ratings_count: "27127"
small_image_url: https://s.gr-assets.com/assets/nophoto/book/50x75-a91bf249278a81aabab721ef782c4a74.png
text_reviews_count: "96"
title: 'Zoe''s Tale (Old Man's War, #4)'
title_without_series: Zoe's Tale
uri: kca://book/amzn1.gr.book.v1.U6u3FgU-gmF98g6PHe67fw
work: |2
    18280032
    kca://work/amzn1.gr.work.v1.xOA6hSv2pTisB00my8yoHQ
wash goodreads >
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)
true, the whole thing, to try make you feel what I felt: the joy and terror and
uncertainty, panic and wonder, despair and hope. Everything that happened, bringing
us to Earth, and Earth out of its captivity. All through my eyes.<br /><br />It's
a story you know. But you don't know it all.'
edition_information: Reprint Edition
id: "8156551"
image_url: https://s.gr-assets.com/assets/nophoto/book/111x148-bcc042a9c91a29c1d680899eff700a03.png
link: https://www.goodreads.com/book/show/8156551-zoe-s-tale
num_pages: "420"
publication_day: "1"
publication_month: "4"
publication_year: "2010"
published: "2010"
publisher: Tor Books
ratings_count: "27127"
small_image_url: https://s.gr-assets.com/assets/nophoto/book/50x75-a91bf249278a81aabab721ef782c4a74.png
text_reviews_count: "96"
title: 'Zoe''s Tale (Old Man's War, #4)'
title_without_series: Zoe's Tale
uri: kca://book/amzn1.gr.book.v1.U6u3FgU-gmF98g6PHe67fw
work: |2
    18280032
    kca://work/amzn1.gr.work.v1.xOA6hSv2pTisB00my8yoHQ
wash goodreads > ls currently-reading
Death by Food Pyramid: How Shoddy Science, Sketchy Politics and Shady Special Interests Have Ruined Our Health
Jonathan Strange & Mr Norrell
Parenting from the Inside Out: How a Deeper Self-Understanding Can Help You Raise Children Who Thrive
The Design of Everyday Things
The Dragon's Path (The Dagger and the Coin, #1)
The Human Side of Agile - How to Help Your Team Deliver
The New Father: A Dad's Guide to the First Year (New Father Series)
The Watchman's Rattle: A New Way to Understand Complexity, Collapse, and Correction
wash goodreads >
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (zsh)
a story you know. But you don't know it all.'
edition_information: Reprint Edition
id: "8156551"
image_url: https://s.gr-assets.com/assets/nophoto/book/111x148-bcc042a9c91a29c1d680899eff700a03.png
link: https://www.goodreads.com/book/show/8156551-zoe-s-tale
num_pages: "420"
publication_day: "1"
publication_month: "4"
publication_year: "2010"
published: "2010"
publisher: Tor Books
ratings_count: "27127"
small_image_url: https://s.gr-assets.com/assets/nophoto/book/50x75-a91bf249278a81aabab721ef782c4a74.png
text_reviews_count: "96"
title: 'Zoe''s Tale (Old Man's War, #4)'
title_without_series: Zoe's Tale
uri: kca://book/amzn1.gr.book.v1.U6u3FgU-gmF98g6PHe67fw
work: |2
    18280032
    kca://work/amzn1.gr.work.v1.xOA6hSv2pTisB00my8yoHQ
wash goodreads > ls .currently-reading
Death by Food Pyramid: How Shoddy Science, Sketchy Politics and Shady Special Interests Have Ruined Our Health
Jonathan Strange & Mr Norrell
Parenting from the Inside Out: How a Deeper Self-Understanding Can Help You Raise Children Who Thrive
The Design of Everyday Things
The Dragon's Path (The Dagger and the Coin, #1)
The Human Side of Agile - How to Help Your Team Deliver
The New Father: A Dad's Guide to the First Year (New Father Series)
The Watchman's Rattle: A New Way to Understand Complexity, Collapse, and Correction
wash goodreads > exit
Goodbye!
-> cd puppetlabs/washreads
~/puppetlabs/washreads > master
```



SCREEN SHARE

The screenshot shows a Mac desktop with a dark-themed window titled "iTerm2". Inside the window, a terminal session is open with a Ruby script named "Server.nxp". The script is designed to interact with the Goodreads API using the Wash command-line interface. The code includes requirements for OAuth libraries and handles authentication by requesting an access token if none is found in a local cache file.

```
#!/usr/bin/env ruby
# A script for exploring your Goodreads library with Wash
#
# Requirements:
# - `gem install oauth nokogiri --user-install`
# Note that BitBar runs without shell setup, so it will likely use the system ruby.
# - A developer key/secret from https://www.goodreads.com/api/keys
# stored in GOODREADS_KEY, GOODREADS_SECRET environment variables.

require 'json'
require 'oauth'
require 'nokogiri'
require 'set'

# Step 1: figure out authentication
consumer = OAuth::Consumer.new(ENV['GOODREADS_KEY'],
                                ENV['GOODREADS_SECRET'],
                                site: 'https://www.goodreads.com')

# If we don't have an access token, request one.
# TODO: module-specific configuration of where to cache it.
# Will need to move this to init, and pass through as plugin state.
TOKEN_CACHE = File.join(Dir.home, '.puppetlabs', 'wash', 'goodreads.token')
begin
  token_text = File.read(TOKEN_CACHE)
  obj = JSON.parse(token_text)
  @access_token = OAuth::AccessToken.new(consumer, obj['token'], obj['secret'])
rescue StandardError
  request_token = consumer.get_request_token
  puts "Please authorize at #{request_token.authorize_url}, then press enter once done"
  gets
  @access_token = request_token.get_access_token
end
"goodreads" 115L, 4161C
```

7,1

Top



SCREEN SHARE

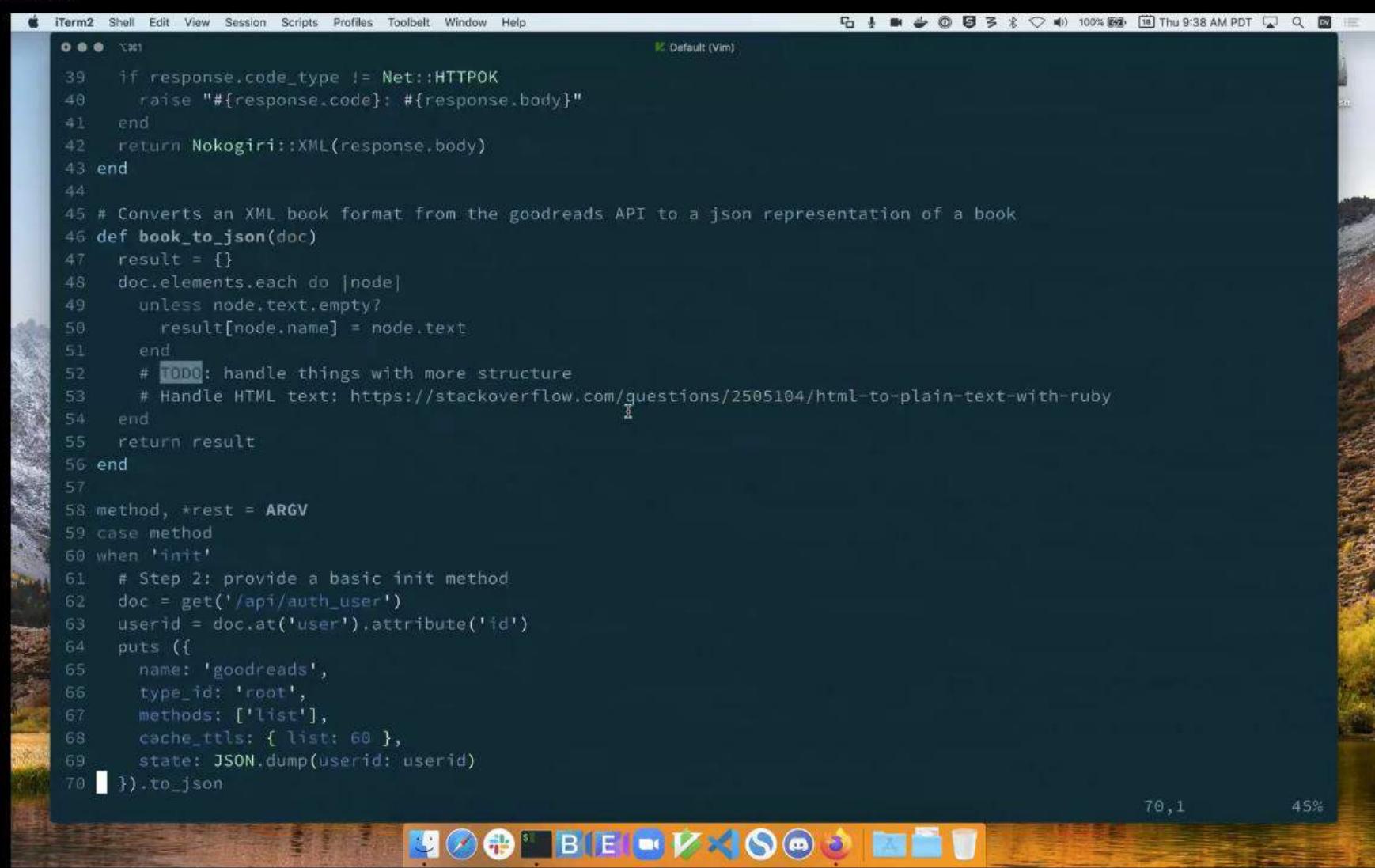
```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (Vim)
31 gets
32 @access_token = request_token.get_access_token
33 File.write(TOKEN_CACHE, { token: @access_token.token, secret: @access_token.secret }.to_json)
34 end
35
36 # Returns an XML response body on OK status, else raises an error
37 def get(path)
38   response = @access_token.get(path)
39   if response.code_type != Net::HTTPOK
40     raise "#{response.code}: #{response.body}"
41   end
42   return Nokogiri::XML(response.body)
43 end
44
45 # Converts an XML book format from the goodreads API to a json representation of a book
46 def book_to_json(doc)
47   result = {}
48   doc.elements.each do |node|
49     unless node.text.empty?
50       result[node.name] = node.text
51     end
52     # TODO: handle things with more structure
53     # Handle HTML text: https://stackoverflow.com/questions/2505104/html-to-plain-text-with-ruby
54   end
55   return result
56 end
57
58 method, *rest = ARGV
59 case method
60 when 'init'
61   # Step 2: provide a basic init method
62   doc = get('/api/auth_user')
```

38,1

36%



SCREEN SHARE



```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help
Default (Vim)

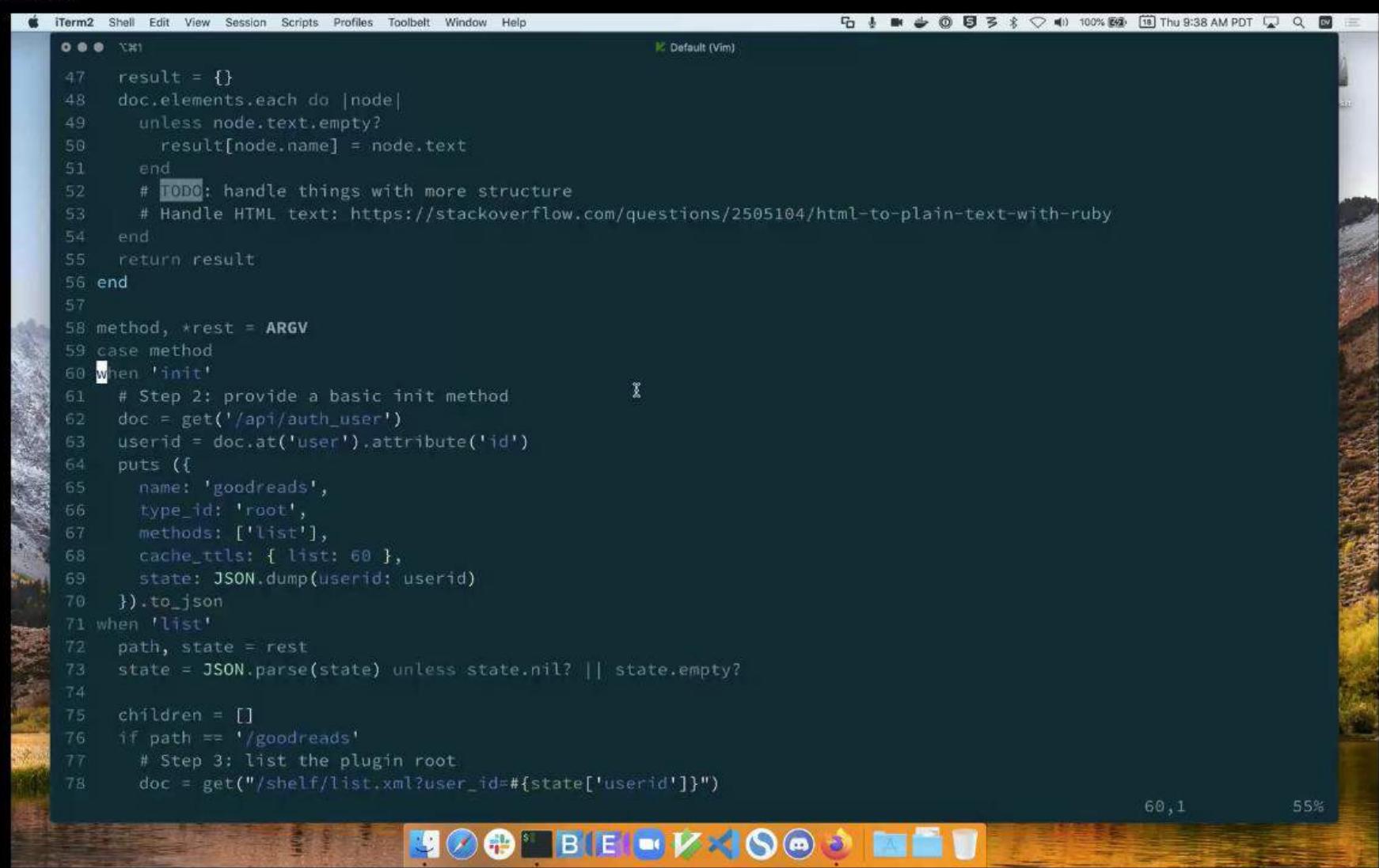
39 if response.code_type != Net::HTTPOK
40   raise "#{response.code}: #{response.body}"
41 end
42 return Nokogiri::XML(response.body)
43 end
44
45 # Converts an XML book format from the goodreads API to a json representation of a book
46 def book_to_json(doc)
47   result = {}
48   doc.elements.each do |node|
49     unless node.text.empty?
50       result[node.name] = node.text
51     end
52     # TODO: handle things with more structure
53     # Handle HTML text: https://stackoverflow.com/questions/2505104/html-to-plain-text-with-ruby
54   end
55   return result
56 end
57
58 method, *rest = ARGV
59 case method
60 when 'init'
61   # Step 2: provide a basic init method
62   doc = get('/api/auth_user')
63   userid = doc.at('user').attribute('id')
64   puts ({
65     name: 'goodreads',
66     type_id: 'root',
67     methods: ['list'],
68     cache_ttls: { list: 60 },
69     state: JSON.dump(userid: userid)
70   }).to_json
70 █
```

70,1

45%



SCREEN SHARE



The screenshot shows a terminal window titled "iTerm2" with a tab labeled "Default (Vim)". The window contains a Ruby script with line numbers from 47 to 78. The script defines a method `handle_html` that processes an XML document to extract text from non-empty nodes. It then handles command-line arguments to either initialize a user or list items. The code uses `HTTParty` to interact with a REST API, specifically fetching user information and listing shelf items.

```
47 result = {}
48 doc.elements.each do |node|
49   unless node.text.empty?
50     result[node.name] = node.text
51   end
52   # TODO: handle things with more structure
53   # Handle HTML text: https://stackoverflow.com/questions/2505104/html-to-plain-text-with-ruby
54 end
55 return result
56 end
57
58 method, *rest = ARGV
59 case method
60 when 'init'
61   # Step 2: provide a basic init method
62   doc = get('/api/auth_user')
63   userid = doc.at('user').attribute('id')
64   puts ({
65     name: 'goodreads',
66     type_id: 'root',
67     methods: ['list'],
68     cache_ttls: { list: 60 },
69     state: JSON.dump(userid: userid)
70   }).to_json
71 when 'list'
72   path, state = rest
73   state = JSON.parse(state) unless state.nil? || state.empty?
74
75   children = []
76   if path == '/goodreads'
77     # Step 3: list the plugin root
78     doc = get("/shelf/list.xml?user_id=#{state['userid']}")
```

60,1

55%



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (Vim)
63   userid = doc.at('user').attribute('id')
64   puts ({
65     name: 'goodreads',
66     type_id: 'root',
67     methods: ['list'],
68     cache_ttls: { list: 60 },
69     state: JSON.dump(userid: userid)
70   }).to_json
71 when 'list'
72   path, state = rest
73   state = JSON.parse(state) unless state.nil? || state.empty?
74
75   children = []
76   if path == '/goodreads'
77     # Step 3: list the plugin root
78     doc = get("/shelf/list.xml?user_id=#{state['userid']}")
79     doc.search('shelves user_shelf').each do |bookshelf|
80       name = bookshelf.at('name').content
81       meta = {}
82       bookshelf.elements.each do |node|
83         next if node.name == 'name'
84         meta[node.name] = node.content
85       end
86       # Step 4: add state we can use to keep track of what we're interacting with. TODO: use Oj for object serialization?
87       count = bookshelf.at('book_count').content.to_i
88       shelf = { type: 'shelf', name: name, userid: state['userid'], count: count }
89       children << { name: name, type_id: 'shelf', methods: ['list'], partial_metadata: meta, state: JSON.dump(shelf) }
90     end
91   elsif state['type'] == 'shelf'
92     pages = (state['count']+199) / 200
72,1 72%
```



SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (Vim)
73 state = JSON.parse(state) unless state.nil? || state.empty?
74
75 children = []
76 if path == '/goodreads'
77   # Step 3: list the plugin root
78   doc = get("/shelf/list.xml?user_id=#{state['userid']}")
79   doc.search('shelves user_shelf').each do |bookshelf|
80     name = bookshelf.at('name').content
81     meta = {}
82     bookshelf.elements.each do |node|
83       next if node.name == 'name'
84       meta[node.name] = node.content
85     end
86     # Step 4: add state we can use to keep track of what we're interacting with. TODO: use Oj for object serialization?
87     count = bookshelf.at('book_count').content.to_i
88     shelf = { type: 'shelf', name: name, userid: state['userid'], count: count }
89     children << { name: name, type_id: 'shelf', methods: ['list'], partial_metadata: meta, state: JSON.dump(shelf) }
90   end
91 elsif state['type'] == 'shelf'
92   pages = (state['count']+199) / 200
93   # Several books may share the same title. For now, ensure we only add each title once.
94   titles = {}
95   pages.times do |idx|
96     # Step 5: list books on a shelf.
97     STDERR.puts "Loading page #{idx+1} / #{pages} for shelf #{state['name']}"
98     doc = get("/review/list/#{$state['userid']}.xml?v=2&shelf=#{state['name']}&page=#{idx+1}&per_page=200")
99     doc.search('reviews review').each do |review|
100       name = review.at('book title').content
101       meta = book_to_json(review.at('book'))
102       if titles.include?(name)
```

91,1

84%



SCREEN SHARE

iTerm2 Shell Edit View Session Scripts Profiles Toolbar Window Help

Default (Vim)

```
80     name = bookshelf.at('name').content
81     meta = {}
82     bookshelf.elements.each do |node|
83       next if node.name == 'name'
84       meta[node.name] = node.content
85     end
86     # Step 4: add state we can use to keep track of what we're interacting with. TODO: use Obj for object serialization?
87     count = bookshelf.at('book_count').content.to_i
88     shelf = { type: 'shelf', name: name, userid: state['userid'], count: count }
89     children << { name: name, type_id: 'shelf', methods: ['list'], partial_metadata: meta, state: JSON.dump(shelf) }
90   end
91 elsif state['type'] == 'shelf'
92   pages = (state['count']+199) / 200
93   # Several books may share the same title. For now, ensure we only add each title once.
94   titles = {}
95   pages.times do |idx|
96     # Step 5: list books on a shelf.
97     STDERR.puts "Loading page #{idx+1} / #{pages} for shelf #{state['name']}"
98     doc = get("/review/list/#{$state['userid']}.xml?v=2&shelf=#{state['name']}&page=#{idx+1}&per_page=200")
99     doc.search('reviews review').each do |review|
100       name = review.at('book title').content
101       meta = book_to_json(review.at('book'))
102       if titles.include?(name)
103         STDERR.puts "Found duplicate title #{name}:\n#{titles[name]}\n#{meta}"
104       else
105         titles[name] = meta
106         children << { name: name, type_id: 'book', methods: [], partial_metadata: meta }
107       end
108     end
109   end
110 end
```

SCREEN SHARE

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Default (Vim)
86      # Step 4: add state we can use to keep track of what we're interacting with. TODO: use Oj for object serialization?
87      count = bookshelf.at('book_count').content.to_i
88      shelf = { type: 'shelf', name: name, userid: state['userid'], count: count }
89      children << { name: name, type_id: 'shelf', methods: ['list'], partial_metadata: meta, state: JSON.dump(shelf) }
90    end
91  elsif state['type'] == 'shelf'
92    pages = (state['count']+199) / 200
93    # Several books may share the same title. For now, ensure we only add each title once.
94    titles = {}
95    pages.times do |idx|
96      # Step 5: list books on a shelf.
97      STDERR.puts "Loading page #{idx+1} / #{pages} for shelf #{state['name']}"
98      doc = get("/review/list/#{state['userid']}.xml?v=2&shelf=#{state['name']}&page=#{idx+1}&per_page=200")
99      doc.search('reviews review').each do |review|
100        name = review.at('book title').content
101        meta = book_to_json(review.at('book'))
102        if titles.include?(name)
103          STDERR.puts "Found duplicate title #{name}:#{titles[name]}\n#{meta}"
104        else
105          titles[name] = meta
106          children << { name: name, type_id: 'book', methods: [], partial_metadata: meta }
107        end
108      end
109    end
110  end
111  puts children.sort_by {|obj| obj['name']}.to_json
112 else
113   STDERR.puts "Unknown method invoked"
114   exit 1
115 end
```



SCREEN SHARE

```
#!/usr/bin/env ruby
# frozen_string_literal: true

require 'wash'

# For now we're going to mock out the plugin interface. It requires more setup.
# to get PAL/PuppetDB working.
class Plugin
  attr_accessor :plugin_hooks
  def initialize
    @plugin_hooks = {}
  end

  def reference?(_input)
    false
  end

  def resolve_references(ref)
    ref
  end

  def resolve_top_level_references(ref)
    ref
  end
end

class Boltwash < Wash::Entry
  label 'bolt'
  is_singleton
  parent_of 'Group'
  description <>-DESC
  A plugin for Puppet Bolt's inventory. You can see target configuration in entry
  metadata. You can SSH to targets and view their filesystems in 'fs' directories.
  DESC

  def init(config)
    require 'bolt/inventory'
    boltdir = config[:dir] ? Bolt::BoltDir.new(config[:dir]) : Bolt::BoltDir.default_boltdir
    bolt_config = Bolt::Config.from_boltdir(boltdir)
    @inventory = Bolt::Inventory.from_config(bolt_config, Plugin.new)
    prefetch :list
  end

  def list
    groups = @inventory.group_names
    groups.map { |group| Group.new(@inventory, group) }
  end
end
```



SCREEN SHARE

```
Open Source Summit + Email · pull · boltwash/bolt.rb at master · pull · + https://github.com/puppetlabs/boltwash/blob/master/bolt.rb
25
26
27 class Boltwash < Wash::Entry
28   label 'bolt'
29   is_singleton
30   parent_of 'Group'
31   description <><DESC
32     A plugin for Puppet Bolt's inventory. You can see target configuration in entry
33     metadata. You can SSH to targets and view their filesystems in 'fs' directories.
34
35     All groups are shown in this directory, with the special 'all' group including
36     all targets in the inventory.
37   DESC
38
39   def init(config)
40     require 'bolt/inventory'
41     boltdir = config[:dir] ? Bolt::BoltDir.new(config[:dir]) : Bolt::BoltDir.default_boltdir
42     bolt_config = Bolt::Config.from_boltdir(boltdir)
43     @inventory = Bolt::Inventory.from_config(bolt_config, Plugin.new)
44     prefetch :list
45   end
46
47   def list
48     groups = @inventory.group_names
49     groups.map { |group| Group.new(@inventory, group) }
50   end
51 end
52
53 class Group < Wash::Entry
54   label 'group'
55   parent_of 'Target'
56   description <><DESC
57     This is a group. Listing it shows all targets in the group.
58   DESC
59
60   def initialize(inventory, name)
61     @inventory = inventory
62     @name = name
63     prefetch :list
64   end
65
66   def list
67     targets = @inventory.get_targets(@name)
68     targets.map { |target| Target.new(target) }
69   end
70 end
71
72 def get_login_shell(target)
73   # Bolt's inventory defines a shell as a feature. Some transports provide
```



SCREEN SHARE

```
▶ Open Source Summit + Ethical Hacking · pull · boltwash/bolt.rb at master · puppetlabs/boltwash/blob/master/bolt.rb
https://github.com/puppetlabs/boltwash/blob/master/bolt.rb
...
24     end
25   end
26
27   class Boltwash < Wash::Entry
28     label 'bolt'
29     is_singleton
30     parent_of 'Group'
31     description <><DESC>
32       A plugin for Puppet Bolt's inventory. You can see target configuration in entry
33       metadata. You can SSH to targets and view their filesystems in 'fs' directories.
34
35       All groups are shown in this directory, with the special 'all' group including
36       all targets in the inventory.
37     DESC
38
39     def init(config)
40       require 'bolt/inventory'
41       boltdir = config[:dir] ? Bolt::BoltDir.new(config[:dir]) : Bolt::BoltDir.default_boltdir
42       bolt_config = Bolt::Config.from_boltdir(boltdir)
43       @inventory = Bolt::Inventory.from_config(bolt_config, Plugin.new)
44       prefetch :list
45     end
46
47     def list
48       groups = @inventory.group_names
49       groups.map { |group| Group.new(@inventory, group) }
50     end
51   end
52
53   class Group < Wash::Entry
54     label 'group'
55     parent_of 'Target'
56     description <><DESC>
57       This is a group. Listing it shows all targets in the group.
58     DESC
59
60     def initialize(inventory, name)
61       @inventory = inventory
62       @name = name
63       prefetch :list
64     end
65
66     def list
67       targets = @inventory.get_targets(@name)
68       targets.map { |target| Target.new(target) }
69     end
70   end
71
72   def get_login_shell(target)
73 
```





VIDEO



SLIDES

Reflections

ABSTRACT

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.

22



**VIDEO****ABSTRACT**

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.

SLIDES

- Macintosh
- Network
- OSXFUSE...me 0 (wash) ▲
- aws
- bolt
- docker
- gcp
- kubernetes
- local
- docker-desktop
- docker-for-desktop
- gke_shared-k...ared-k8s-dev
- gke_shared-k...red-k8s-prod
- gke_shared-k...ed-k8s-stage
- kubernetes-a...@kubernetes

SSHFS

23



**VIDEO****SLIDES**

```
wash
welcome to Wash! Try 'docs .'
wash .> wash -h
when invoked without arguments, enters a Wash shell. Starts the Wash daemon,
then starts your system shell with shortcuts configured for wash subcommands.

Usage:
wash [script] [flags]
wash [command]

Available Commands:
clear      Clears the cache at the specified paths, or current directory if not specified
delete     Deletes the entries at the specified paths
docs       Displays the entry's documentation
exec      Executes the given command on the indicated target
find      Prints out all entries that satisfy the given expression
help       Help about any command
history   Prints the wash command history, or journal of a particular item
info      Wash's version of 'stat'. Prints the entries' info at the specified paths
ls        Lists the children of the specified paths, or current directory if not specified
meta     Prints the metadata of the given entries
ps        Lists the processes running on the indicated compute instances
server   Sets up the Wash daemon (API and FUSE servers)
signal   Sends the specified signal to the entries at the specified paths
tree     Displays the entry's tree (schema-tree)
tail     Displays new output of files or resources with the stream action
validate Validates an external plugin

Flags:
-c, --command string      Run the supplied string and exit
--config-file string      Set the config file's location (default "./puppetlabs/wash/wash.yaml")
--cpuprofile string       Write cpu profile to file
-h, --help                 help for wash
--logfile string          Set the log file's location. Defaults to stdout
--loglevel string          Set the logging level (default "warn")
--verify-install           Verifies a given Wash installation
--version                  Print the Wash version

Use "wash [command] --help" for more information about a command.
wash .> exit
goodbye!
```

a shell

24

ABSTRACT

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.





VIDEO



ABSTRACT

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.

SLIDES

```
rng-trszc rng 10.38.0.15 - - [08/Mar/2019 15:35:37] "GET /32 HTTP/1.1" 200 -
rng-trszc rng 10.40.0.17 - - [08/Mar/2019 15:35:37] "GET /32 HTTP/1.1" 200 -
rng-c5c74 rng 10.38.0.11 - - [08/Mar/2019 15:35:37] "GET /32 HTTP/1.1" 200 -
rng-trszc rng 10.40.0.16 - - [08/Mar/2019 15:35:37] "GET /32 HTTP/1.1" 200 -
rng-5c58f7cf6-28vqk rng 10.38.0.16 - - [08/Mar/2019 15:35:37] "GET /32 HTTP/1.1" 200 -
rng-f7w84 rng 10.44.0.17 - - [08/Mar/2019 15:35:37] "GET /32 HTTP/1.1" 200 -
rng-c5c74 rng 10.44.0.15 - - [08/Mar/2019 15:35:38] "GET /32 HTTP/1.1" 200 -
rng-trszc rng 10.40.0.18 - - [08/Mar/2019 15:35:38] "GET /32 HTTP/1.1" 200 -
rng-c5c74 rng 10.38.0.14 - - [08/Mar/2019 15:35:38] "GET /32 HTTP/1.1" 200 -
rng-f7w84 rng 10.44.0.16 - - [08/Mar/2019 15:35:38] "GET /32 HTTP/1.1" 200 -
rng-c5c74 rng 10.38.0.15 - - [08/Mar/2019 15:35:38] "GET /32 HTTP/1.1" 200 -
rng-f7w84 rng 10.40.0.17 - - [08/Mar/2019 15:35:38] "GET /32 HTTP/1.1" 200 -
rng-trszc rng 10.38.0.11 - - [08/Mar/2019 15:35:38] "GET /32 HTTP/1.1" 200 -
rng-c5c74 rng 10.40.0.16 - - [08/Mar/2019 15:35:38] "GET /32 HTTP/1.1" 200 -
rng-f7w84 rng 10.38.0.16 - - [08/Mar/2019 15:35:39] "GET /32 HTTP/1.1" 200 -
rng-5c58f7cf6-28vqk rng 10.44.0.17 - - [08/Mar/2019 15:35:39] "GET /32 HTTP/1.1" 200 -
rng-c5c74 rng 10.44.0.15 - - [08/Mar/2019 15:35:39] "GET /32 HTTP/1.1" 200 -
rng-trszc rng 10.40.0.18 - - [08/Mar/2019 15:35:39] "GET /32 HTTP/1.1" 200 -
rng-f7w84 rng 10.38.0.14 - - [08/Mar/2019 15:35:39] "GET /32 HTTP/1.1" 200 -
rng-c5c74 rng 10.44.0.16 - - [08/Mar/2019 15:35:39] "GET /32 HTTP/1.1" 200 -
^C
```

```
[35.180.98.50] (kubernetes-admin@kubernetes:default) docker@node1 ~/container.training/
dockercoints
$ stern --tail 1 r
[0] 0:bash- 1:bash*
```

Stern: <https://youtu.be/-2L1GHTBteY>

"node1" 15:35 08-Mar-19





VIDEO



SLIDES

ABSTRACT

Cloud infrastructure has made it more difficult to understand what's happening across multiple systems, how our systems interact, and even what we're running at this particular moment. More distributed systems have sprung up to help you manage that.

But sometimes setting up those tools is as complicated as the infrastructure you're running. Or they're not flexible in just the way you want. We still spend our days on the command-line, using a suite of tools that do similar things against similar types of infrastructure. But these tools are often inconsistent or inscrutable.

This talk reimagines how we build command-line tools for cloud infrastructure to make them more composable and transferable between systems. I will describe how the Wide-Area Shell (Wash) applies these ideas in designing a new shell experience. I will also share examples of how tools built around this idea can be used in novel ways to help understand your cloud infrastructure.

I've given a similar talk, but this will be longer and highlight more ways the shell design enhances usability.

What could you build?

<https://puppetlabs.github.io/wash/docs/external-plugins#example-plugins>

26





THE LINUX FOUNDATION



VIDEO



SLIDES

Classic Tools for the Modern Age: Introducing Wash, the cloud native shell

Michael Smith, Puppet

Enter question here...

NEW QUESTION

