“The Oregon Trail Deluxe”(1992) by MECC

Contents

whoami

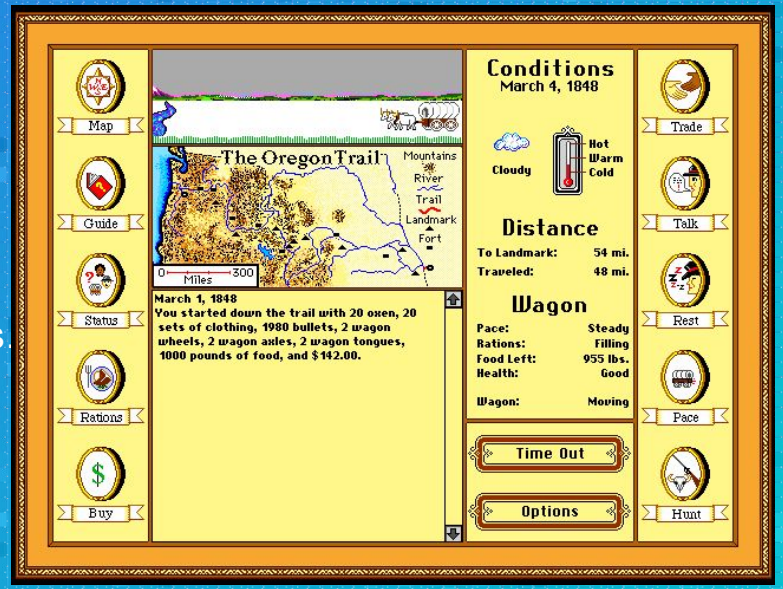
- @joshroppo github.com/ropes
- Data, Platform, and Infrastructure Engineering
  - Cloudy, DevO[o]ps, SRE
- Gopher and Kubernetes User
  - Former Pythonista
- Asking why and thoughtful design
- Dislike 3am pages (firefighting)
- Like stable infrastructure platforms
  - Really dislike reading shiny-new-thing documentation at 3am
- AFK: Mountain biking, Skiing, Climbing Mountains

The Lytics Trailhead

- Customer Data Platform
  - Realtime Web Personalization
  - Web APIs Daily Peak around 3k request/s
  - “Big Data” scale: Petabytes
  - Punching above our weight, finite resources
- Standardized on Go since 0.9
  - Supported by R and Python
- AWS -> Google Cloud Platform[GCE]
  - Performance, consistency, clean tooling for compute commodities.
  - GCP Services: Storage, Pubsub, Bigtable, Bigquery
- “Microservice” when necessary..
  - Three Primary Service Tiers(Miniliths)
    - APIs
    - Workflow Management(Batch Jobs)
    - Event Stream Identity Resolution Processing/ML Classifications

Lytics Trailhead

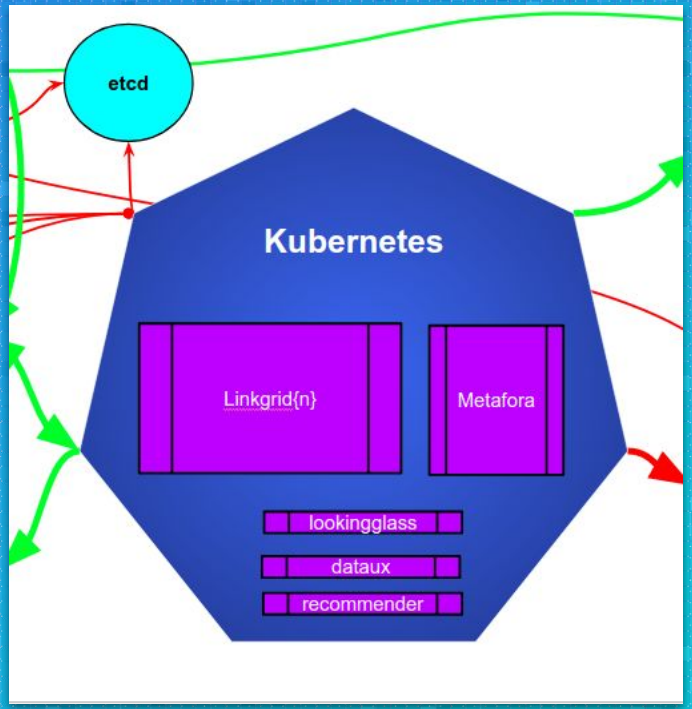
- Saltstack CM
  - Jinja Templatable YAML -> Python -> Bash
  - Powerful and terrifying
- Why Kubernetes?
  - ...this is Kubecon
  - VM management is heavy
    - Building: Packer, OSes, Kernels, Upgrades. single (static)Binary?
    - How big is thy Ops Team?
  - Glad to be on Google Cloud Platform.
    - GKE == Easy Mode!



Loading the Wagon

Goals

- Lytics Applications managed by Kubernetes
  - Everything in Kubernetes(eventually)
- Sustainable Management(ALAP)
  - Roadmap for growing Kubernetes object declarations
    - 1..N ResourceGroups(Deployments)
  - Avoid mountains of static YAML



Loading the Wagon

- Kubernetes Application Prep(Read the Borg Paper)
  - “Container Native” refactor
  - Services/Pods rebalance gracefully
  - No persistent local storage use.
    - GCS, Ceph, NAS, etc
  - HTTP PreStop hook
    - Annoying batch workflows that shouldn’t be killed..
- Application Prep Delayed Kubernetes Deployment

Compute Resource Massacre

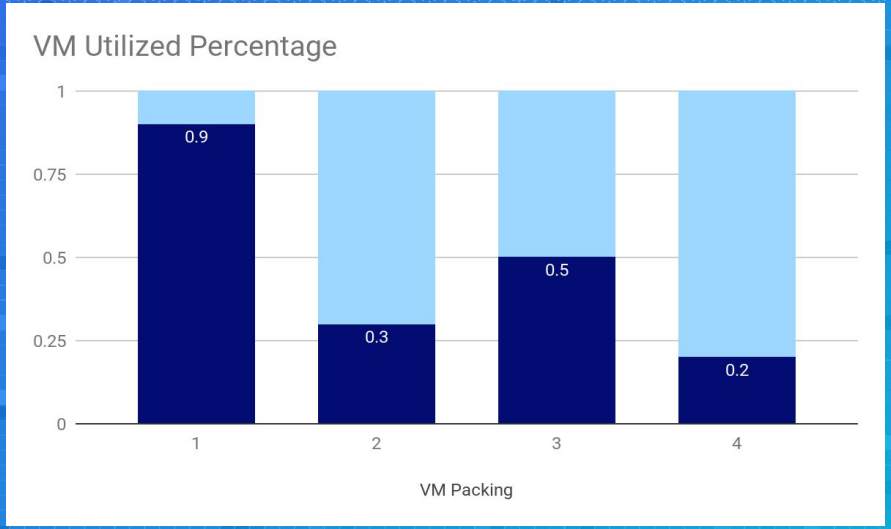
- Heterogeneous Workloads are difficult to bin pack.
- Flat vs Complex workflow schedulers.
- CM Managed VM bin packing →

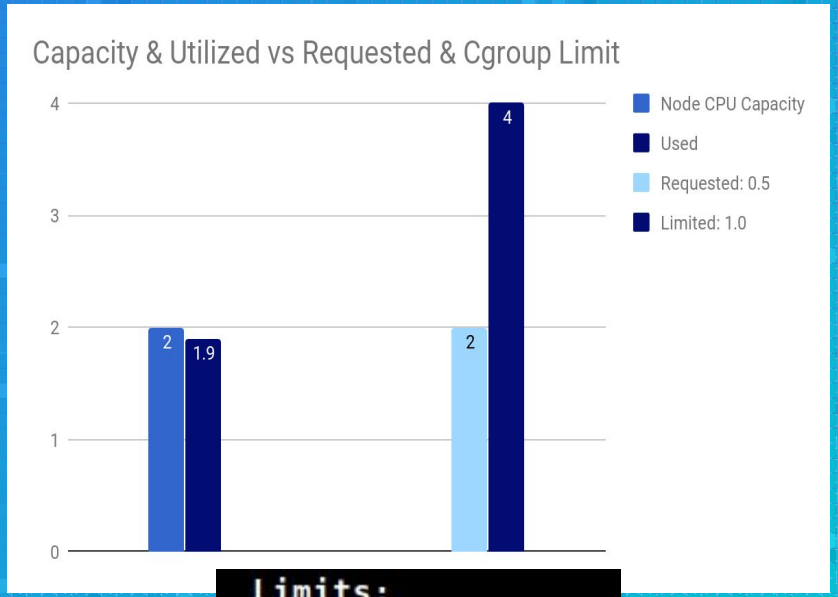
Hunting Outcome:

You shot 2173 pounds of food and used 20 bullets, but were only able to carry 200 pounds of food back.

Compute Resource Massacre

- Container Requests and Limits
  - Allow potential over consumption
  - Scheduler handles Bin Packing
  - “gcloud beta container clusters resize…”



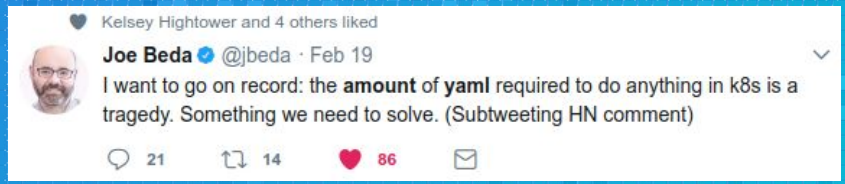



```
Limits:
  cpu:     8
  memory:  26Gi
Requests:
  cpu:     4
  memory:  20Gi
```

On the Trail

YAML (“It’s better than XML”--devopsdays)

- All the tutorials use(d) YAML
- “kubectl apply -f ./tutorial”
  - ^ wat
- Rules and Static Resources: Great!
  - Services, RBAC, Network Policies
- Saltstack Jinja templating in YAML files PTSD
  - Helm; the Kubernetes CM?
    - Smaller scope than Saltstack
      - Simpler Templating
      - Validation more than passing isValidYaml()?
    - Helm seemed more focused on accessibility rather than determinism
    - Good Design, uncertain on execution
- New tools since we started: Ksonnet, Kompose



Go Kubernetes!

client-go

- https://github.com/kubernetes/client-go
- Kubecon[client-go The Good, The Bad, The Ugly]: @LiliCosic https://www.slideshare.net/LiliCosic/clientgo-the-good-the-bad-and-the-ugly
- Why: Initially a test, but provides useful sanity checks
  - Compiling against Kubernetes Source*
  - Static API when Documentation was sporadic
  - CoreOS etcd/Prometheus/(Now Vault!) Operators
- Composibile Data Structures
  - Unit testable; Before sending to Kubernetes

The GoShip logo is an adaptation of the Go gopher created by Renee French under the Creative Commons Attribution 3.0 license.

*: At some revision...

Go Kubernetes!

- Data structure type and Documentation traversal in editors.
- Trivial to return to YAML.
- VS Code Demo!

```
volumes:
- downwardAPI:
    defaultMode: 420
    items:
    - fieldRef:
        apiVersion: v1
        fieldPath: metadata.labels
      path: labels
  name: metadata-labels
```

```
func LioConfigPodVolumes() []v1.Volume {
	return []v1.Volume{
		v1.Volume{
			Name: MetadataLabels,
			VolumeSource: v1.VolumeSource{
				DownwardAPI: &v1.DownwardAPIVolumeSource{
					Items: []v1.DownwardAPIVolumeFile{
						v1.DownwardAPIVolumeFile{
							Path: "labels",
							FieldRef: &v1.ObjectFieldSelector{
								FieldPath: "metadata.labels",
							},
						},
					},
				},
			},
		},
		v1.Volume{
			Name: SecureConfVolumeName,
```

Rigging a CLI

- CLI Tool to create Kubernetes Resources from Go
  - Deployments, DaemonSets, Specific Helper Functions
  - Examples:
    - rigging deployment metaforarunner create --image=gcr.io/… --kubecontext={...}
    - rigging deploymentset linkgrid create --image=gcr.io/… --kubecontext={...}
- kubectl still used :(
  - rigging has limited scope
- Flags can mutate default runtime values
  - Eg: rigging deployment metaforarunner update --cpu=12 --kubecontext={...} (default 8 cpus)
- Hindsight Tips
  - Require Kubeconfig Context selection flag.
  - SemVer binary check before run

Deprecation Dysentery

- Keeping up to date on Kubernetes is hard
  - Project Velocity
  - SIGs, designs to stay aware of. Will a foundation be deprecated in two release cycles?
  - (ReplicationControllers, ThirdPartyResources, ~~PetSets~~)
  - Not the only one(Tiller & Prom-Operator TPR)
- Documentation was unversioned for a long time

http://mentalfloss.com/article/28968/where-are-they-now-diseases-killed-you-oregon-trail

SIGs and Working Groups

Most community activity is organized into Special Interest Groups (SIGs), time bounded Working Groups, and the community meeting.

SIGs follow these guidelines although each of these groups may operate a little differently depending on their needs and workflow.

Each group's material is in its subdirectory in this project.

When the need arises, a new SIG can be created

Master SIG List

Name	Label	Leads	Contact	Meetings
API Machinery	api-machinery	* Daniel Smith, Google * David Eads, Red Hat	* Slack * Mailing List	* Wednesdays at 11:00 PT (Pacific Time) (biweekly)
AWS	aws	* Justin Santa Barbara * Kris Nova, Microsoft * Chris Love * Mackenzie Burnett, Redspread	* Slack * Mailing List	* Fridays at 9:00 PT (Pacific Time) (biweekly)
Apps	apps	* Michelle Noorali, Microsoft * Matt Farina, Samsung SDS * Adnan Abdulhussein, Bitnami	* Slack * Mailing List	* Mondays at 9:00 PT (Pacific Time) (weekly)
Architecture	architecture	* Brian Grant, Google * Jaice Singer DuMars, Microsoft	* Slack * Mailing List	* Thursdays at 15:30 UTC (weekly)
Auth	auth	* Eric Chiang, CoreOS * Jordan Liggitt, Red Hat * David Eads, Red Hat	* Slack * Mailing List	* Wednesdays at 18:00 UTC (biweekly)
Autoscaling	autoscaling	* Marcin Wielgus, Google * Solly Ross, Red Hat	* Slack * Mailing List	* Mondays at 14:00 UTC (biweekly/triweekly)
Azure	azure	* Jason Hansen, Microsoft * Cole Mickens, Microsoft * Jaice Singer DuMars, Microsoft	* Slack * Mailing List	* Wednesdays at 16:00 UTC (biweekly)
Big Data	big-data	* Anirudh Ramanathan, Google * Erik Erlandson, Red Hat	* Slack * Mailing List	* Wednesdays at 17:00 UTC (weekly)
CLI	cli	* Fabiano Franz, Red Hat * Phillip Wittrock, Google * Tony Ado, Alibaba	* Slack * Mailing List	* Wednesdays at 09:00 PT (Pacific Time) (biweekly)
Cluster Lifecycle	cluster-lifecycle	* Luke Marsden, Weave * Joe Beda, Heptio * Robert Bailey, Google * Lucas Käldström, Lucas Labs (occasionally	* Slack * Mailing List	* Tuesdays at 09:00 PT (Pacific Time) (weekly)

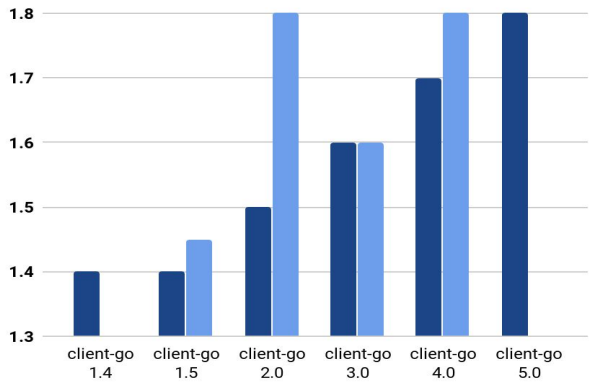
...

Blizzards of Kubernetes

- Surviving the changeset storm
- Client-go codebase matches Kubernetes
  - Brace for changes

+4,103,827 −473,023

- Version Matrix & Compatibility Charts
  - Import path and types were moved(gist)
- Significant improvements made!



Compatibility matrix

	Kubernetes 1.4	Kubernetes 1.5	Kubernetes 1.6	Kubernetes 1.7	Kubernetes 1.8
client-go 1.4	✓	-	-	-	-
client-go 1.5	+	-	-	-	-
client-go 2.0	+-	✓	+-	+-	+-
client-go 3.0	+-	+-	✓	-	+-
client-go 4.0	+-	+-	+-	✓	+-
client-go 5.0	+-	+-	+-	+-	✓
client-go HEAD	+-	+-	+-	+-	+

Blizzards of Kubernetes

Tips

- Spend time on design.
- Vendor vendor vendor! Carefully
  - “dep” works*
- Never mix with other Go projects that use Google Libs and particularly GRPC!
- (Kubernetes Project Velocity cannot be matched)

* careful with RBAC v5.0

Handyman's Corner

Logging

- Logspout
  - DaemonSet fork based on github.com/gliderlabs/logspout
  - Read Docker logs; POST to Elasticsearch Cluster(ELK)
  - 6 months uptime; Multiple K8s Release Upgrades, No changes needed
- Event-logger
  - Collect Kubernetes Events and emit to logging
  - Alert from failure events
- Modified logrus and shimmed for GCP JSON Severity Log Formating

Handyman's Corner

Metrics

- github.com/ropes/Stonecutters
  - Distributed(etcd) UID → Metric alias reuse
  - deploymentpod-87c6bc2f-gssr → deploymentpod-1...N
    - ...Use Prometheus

Interactive Operators

Seth

- Not end goal → Operator
- Provides visibility
- Slack is the communication channel
- Executes commands interactively
- Reports status/results
- Similar to Helm’s Tiller
    - Less sophisticated

Interactive Operators

Poka Yoke
  - Guardrails to limit accidents
  - Validation possibilities
- Staging manager
  - Enables rapid staging deployments
- Hindsight Tips
  - Wrap communication interface(slack…)
  - Write a lexer: Lexical Scanning in Go -- Rob Pike
- E2E Testing DEMO!

Client-go Wrap Up

- Ikea Syndrome? Possibly...
- The client-go path is a lot less bumpy now.
  - Encourage others to explore more.
- GKE is rock solid.(So far)

The Homestead

- Like most risky actions, didn’t go completely to plan, but we made it!
- Clean and scalable resource management delivers.
- Compute scheduling overlap saves cost.
- Tooling only has to interact with one Compute Resource management API.
- Deployments Objects reduce time and concern.

Next: Sim City

Kubernetes is now {N}% more boring!

Service Meshes! Built in Encryption!

(Thanks Istio, Tigera)

Autoscaling & Resizing: $--

Thank You!

freenode: ropes

github.com/ropes

@joshroppo

"I'm pulling for you, we're all in this together!"

Credits

“The Oregon Trail Deluxe” 1992 created by MECC

- Playable at https://archive.org/details/msdos_Oregon_Trail_Deluxe_The_1992
- Read the Borg Paper
- Gist to on Upgrading client-go (your mileage may very)
- client-go example project: github.com/ropes/k8s-trailhead