Cloud Agnostic Design for Fun and Profit

Alexander Meijer
Infrastructure Lead, Corsha

Anusha Iyer
CTO, Corsha

2021 October 13

KubeCon | CloudNativeCon

North America 2021

Welcome!

Alex

Infrastructure Lead at Corsha
First discovered k8s in 2017
Experienced the pain of vendor lock-in at other places

Anusha

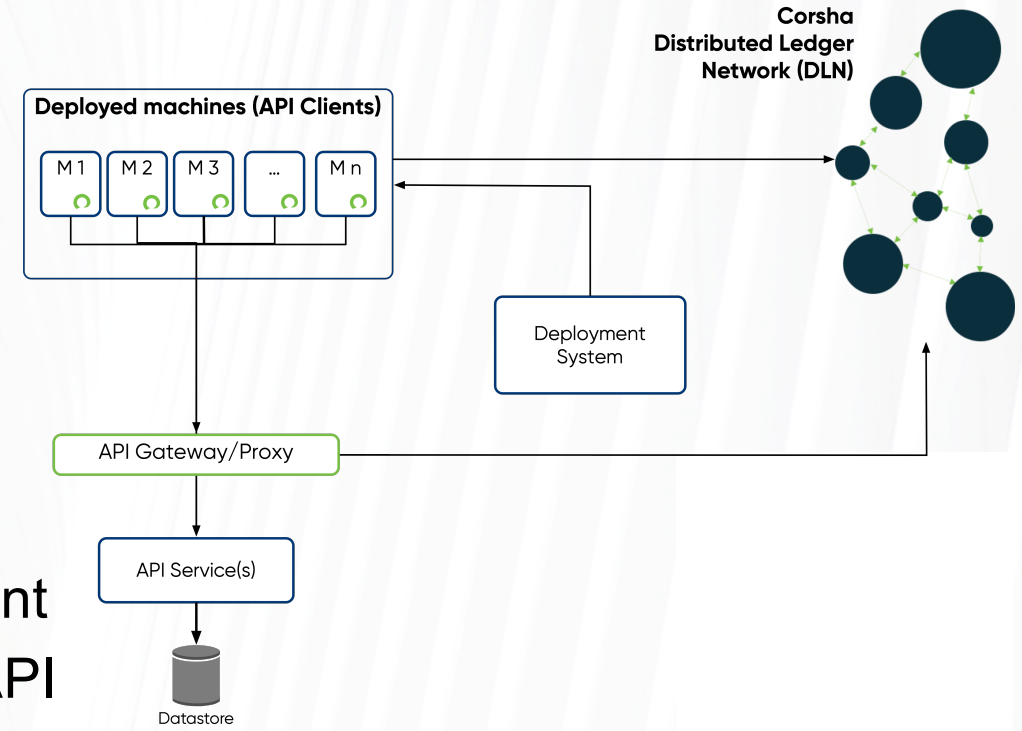
Co-Founder & CTO at Corsha
First discovered k8s from Alex
In love with Infrastructure as Code

Agenda

- Setting the stage: Our app & why we care
- Fundamentals: Helm Charts and Helmfiles
- Economics 101 of Cloud Computing
- Terraforming
- Our Journey
- Scaling these concepts
- Lessons learned
- Where we are going

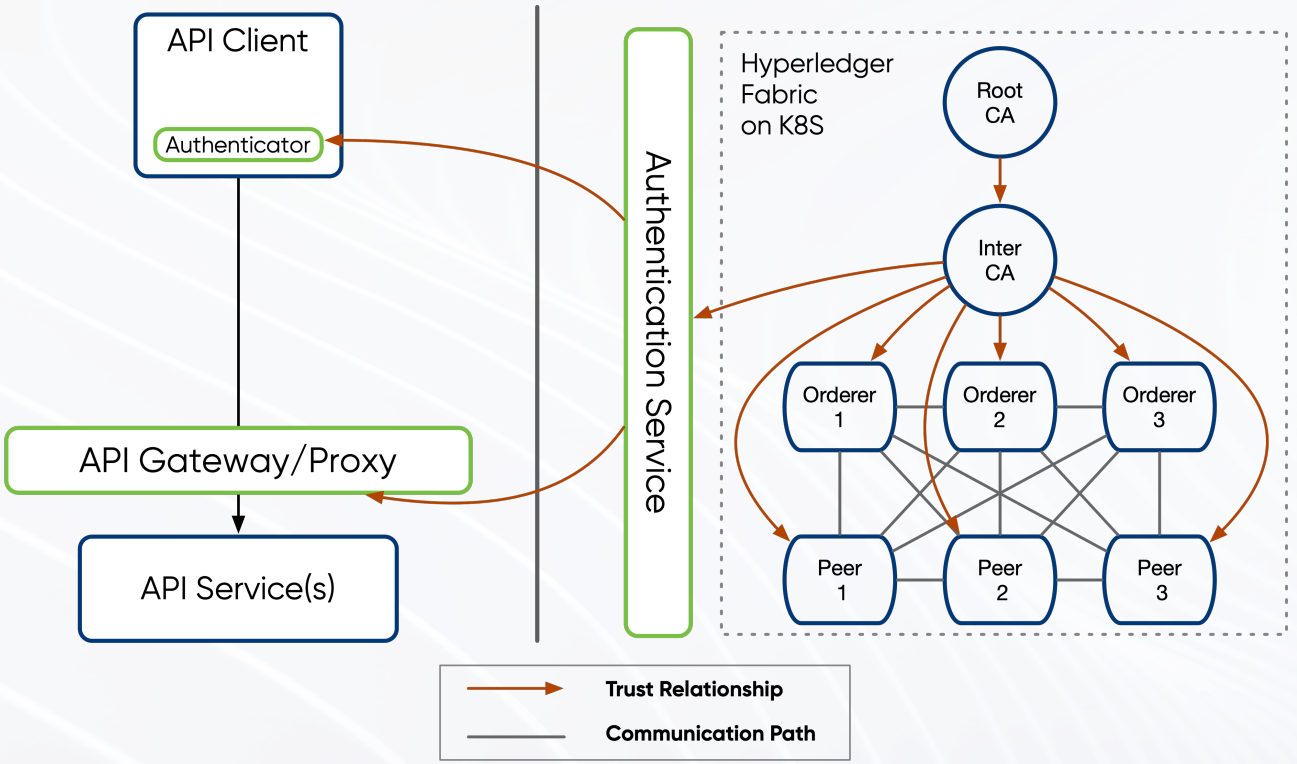
What We Do

- Early stage startup (8 people)
- We automate MFA for API clients
  - Think Google Authenticator
  - RSA tokens but for machines
- We develop a security product
- So we need…
  - High availability
  - SaaS and on-prem deployment
  - Scale to 1000's of real-time API requests per second



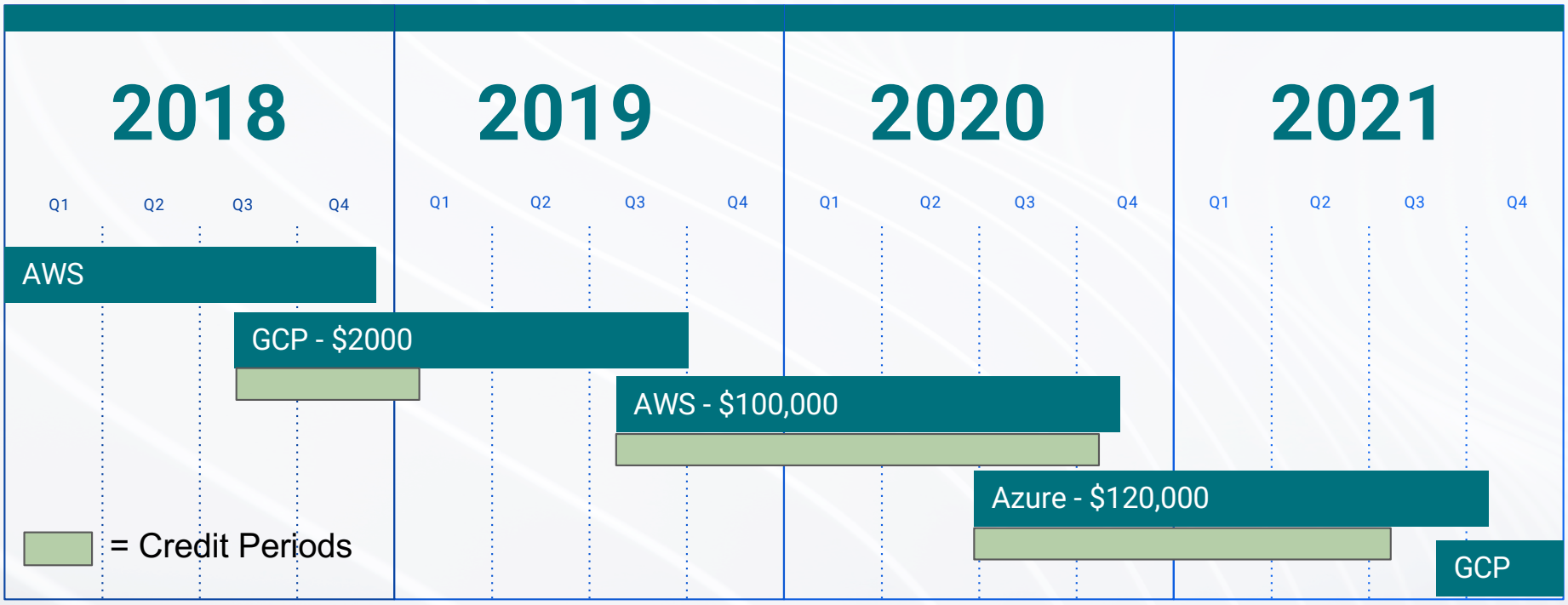
The App...

- Permissioned Distributed Ledger Network (DLN)
  - Persistence tier
- Microservices front the ledger

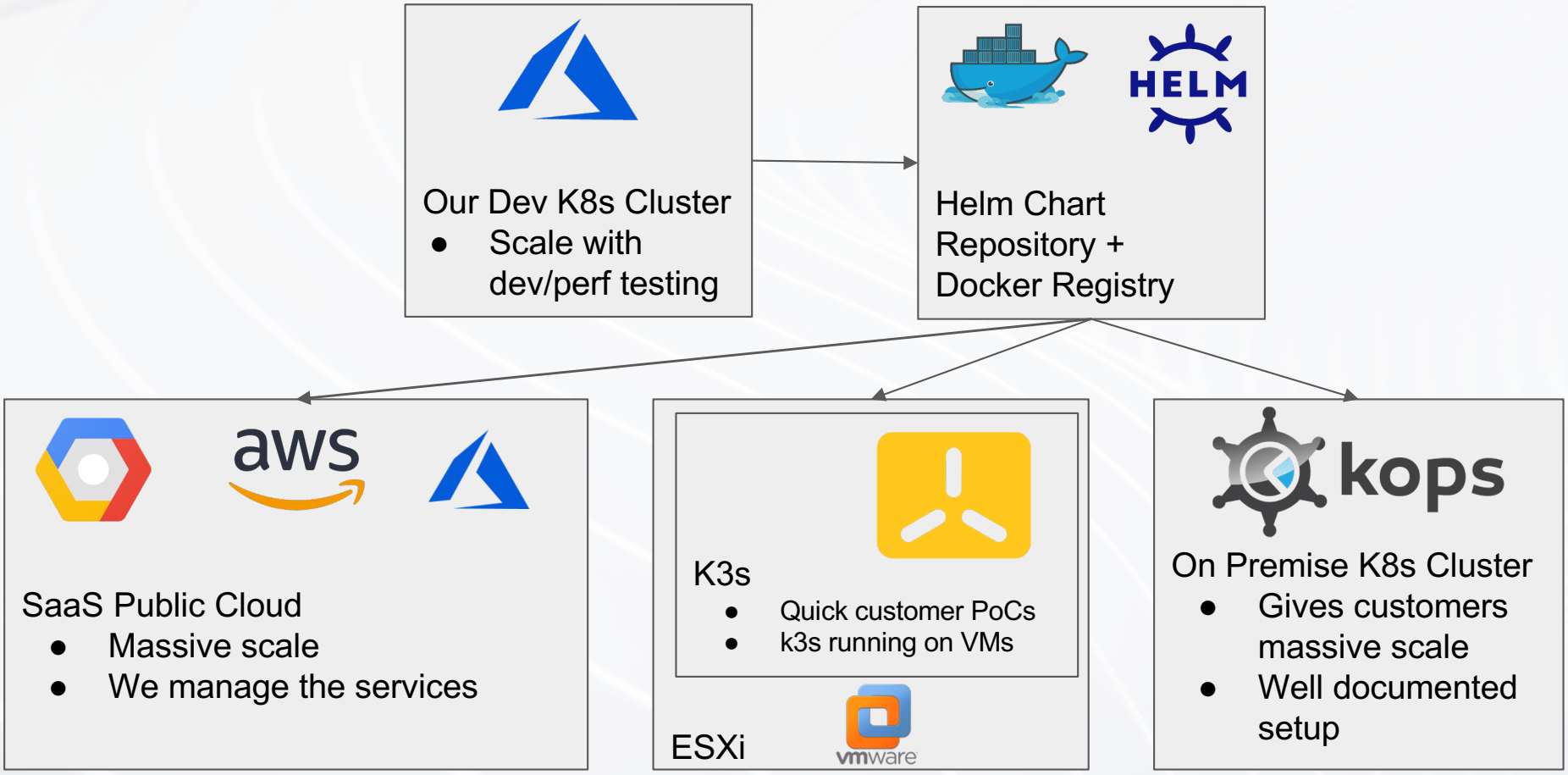


… That Runs For Free

We have run our dev, staging, and production clusters through startup programs -- $222,000 in cloud services credit and counting



The Key is Kubernetes



Form Follows Function

Requirements	
Distributed consensus and high fault tolerance	Use BFT system (e.g., blockchain)
Run across wide range of providers at scale	Use Kubernetes as the indirection layer
Run in air gapped networks	Omit cloud provider services or other SaaS as design constraint
Implement production grade services quickly	Use rich ecosystem of helm charts from artifact hub/stable repos

Case Study: Bitnami Postgres Chart

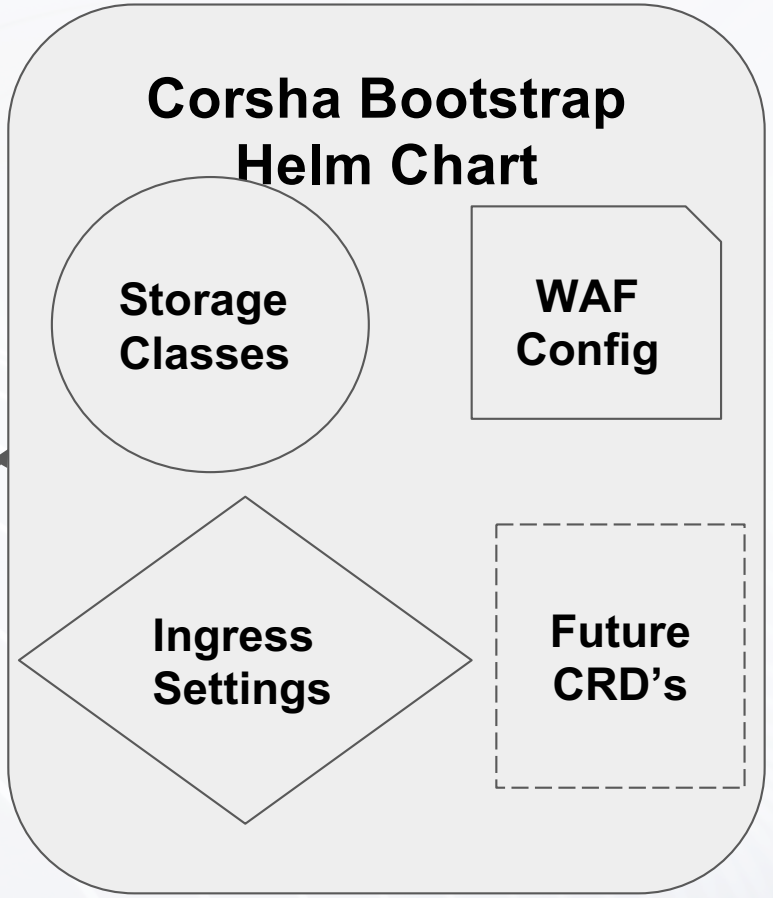
RDS Feature	Bitnami HA Postgres Chart
Easy, managed deployments	`$ helm repo add bitnami https://charts.bitnami.com/bitnami` `$ helm install my-release bitnami/postgresql-ha`
Fast, predictable storage	`global.storageClass`
Backup/ Recovery	`helm install velero vmware-tanzu/velero` [On our Roadmap]
High availability/ Read replicas	`postgresql.replicaCount`
Monitoring/ metrics	`metrics.enabled`
Isolation security	`postgresql.tls.enabled`
Service Level Agreement (SLA)	

Source: (1) https://aws.amazon.com/rds/postgresql/

A Bootstrap Chart

- Concentrate all cloud specific config into this chart
- Level sets clusters

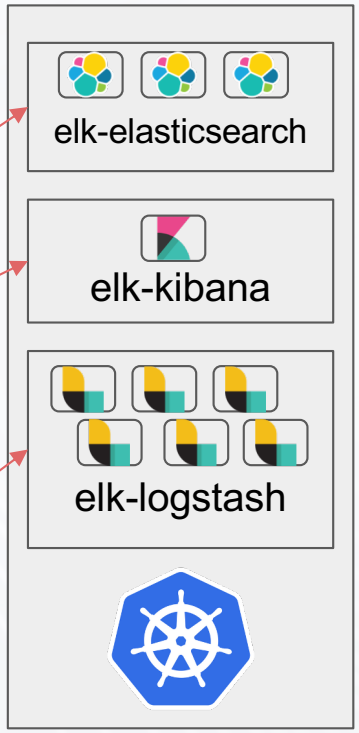
```
{{ if eq .Values.provider "gke"}}
apiVersion: storage.k8s.io/v1
...
provisioner: kubernetes.io/gce-pd
{{- else if eq .Values.provider "aws" -}}
apiVersion: storage.k8s.io/v1
...
provisioner: kubernetes.io/aws-ebs
parameters:
 encrypted: "true"
{{- else if eq .Values.provider "azure" -}}
...
```



Deploying Many Charts With Helmfile

- Cumbersome managing large groups of helm charts
  - **Answer: helmfile (roboll/helmfile)**
- Multiple helmfiles
  - Core services
  - Dev services/CICD
  - Core stack components
- Helmfiles + helm vars (encrypted with sops where necessary) in git

```
repositories:
 - name: elastic
   url: https://helm.elastic.co
releases:
 - name: elk-elasticsearch
   namespace: elk
   version: 1.2.3
   chart: elastic/elasticsearch
   secrets:
     - path/to/encrypted/values/file.yaml
 - name: elk-kibana
   namespace: elk
   version: 4.5.6
   chart: elastic/kibana
 - name: elk-logstash
   namespace: elk
   version: 7.8.9
   chart: elastic/logstash
   values:
     - path/to/values/file.yaml
```



An Emergent Property: Cloud Agnosticism

- The ability to deploy on prem, at scale, with the same exact infrastructure for public clouds
- We can go from 0 to a fully operational dev cluster, staging, prod in a matter of minutes with just a kubeconfig file pointing at a cluster(s)
- All the major cloud providers offer managed k8s (and kops + openstack offer access to many more)
- We are not using anything that we wouldn’t ‘bring with us’ into a new cluster

Cloud Computing Economics 101

- Cloud computing is becoming a commodity → Battle over market share
- Some cloud providers are profitable and defending market share
- Others are willing to take a loss to gain market share
- This competitive environment creates opportunities for cloud users

Hourly Instance* Savings	From GCP	From AWS	From Azure	From DigitalOcean
To GCP		1%	13%	8%
To AWS	1%		15%	7%
To Azure	12%	12%		19%
To DigitalOcean	9%	8%	23%	

* GCP=n2-standard-8 (us-central1, $0.388/hr), AWS=m5.2xlarge (us-east-2, $0.384), Azure=D8s_v3(central-us, $0.44), DigitalOcean=General Purpose Droplet(SFO?, $0.357), retrieved 25 Aug 2021

The Challenge

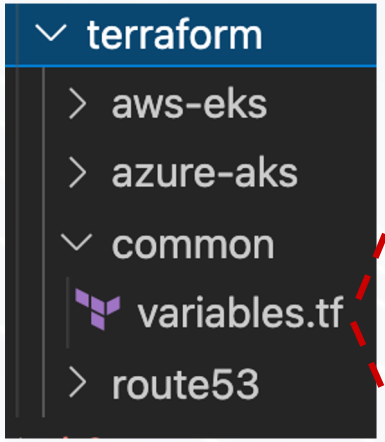
Ok, great, we can save money if we switch cloud providers

- BUT -

- Need to get cloud provider configured to the point of being able to receive k8s API commands
- We are getting there (Fargate et. al), but still no ‘push button’ k8s + worker node deployment
  - Workloads/use cases are just too diverse

Terraform to the Rescue

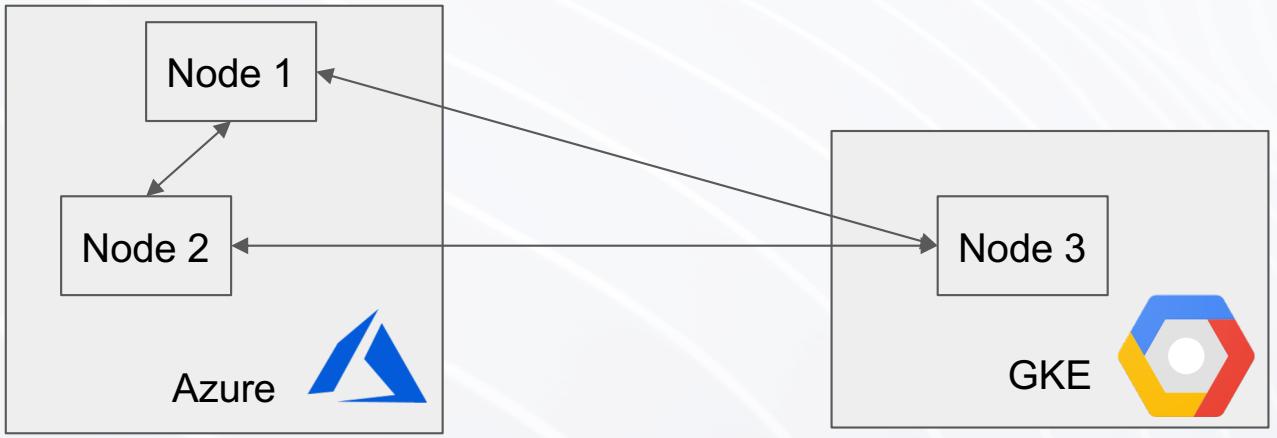
- The key to us going from 0 to deployed in < 2 weeks
- All the cloud providers maintain good terraform modules
- Keep all provider-agnostic config in a ‘common’ module
- Utilize terraform workspaces



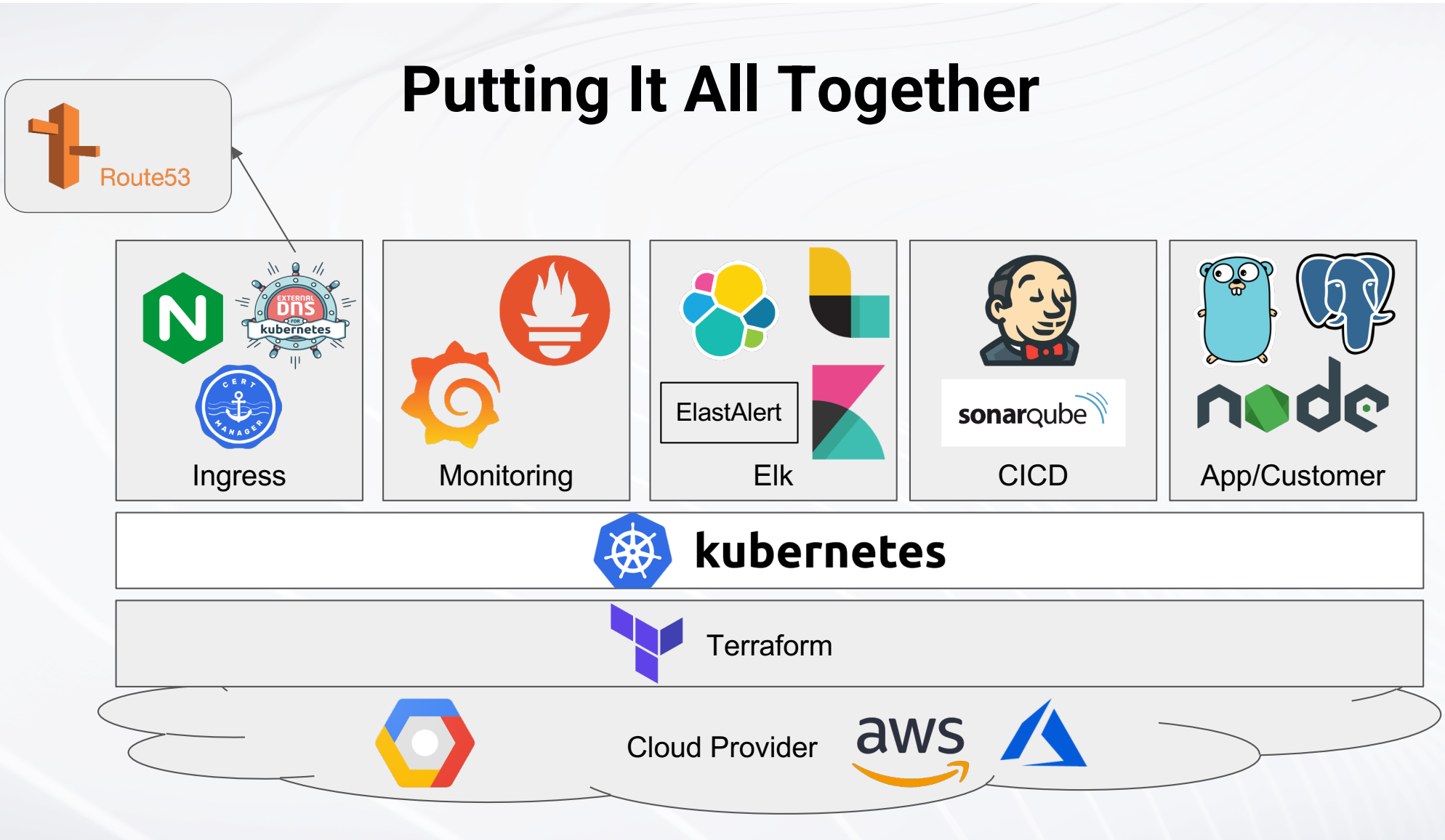

```
output "environment" {
 value = {
   default = {
     max-size = "32"
   }
   dev = {
     ip-whitelist = [
       "1.2.3.4"
     ]
   }
   prod = {
     ip-whitelist = [
       "4.5.6.7"
     ]
   }
 }
}
```

Migrating Prod / DR

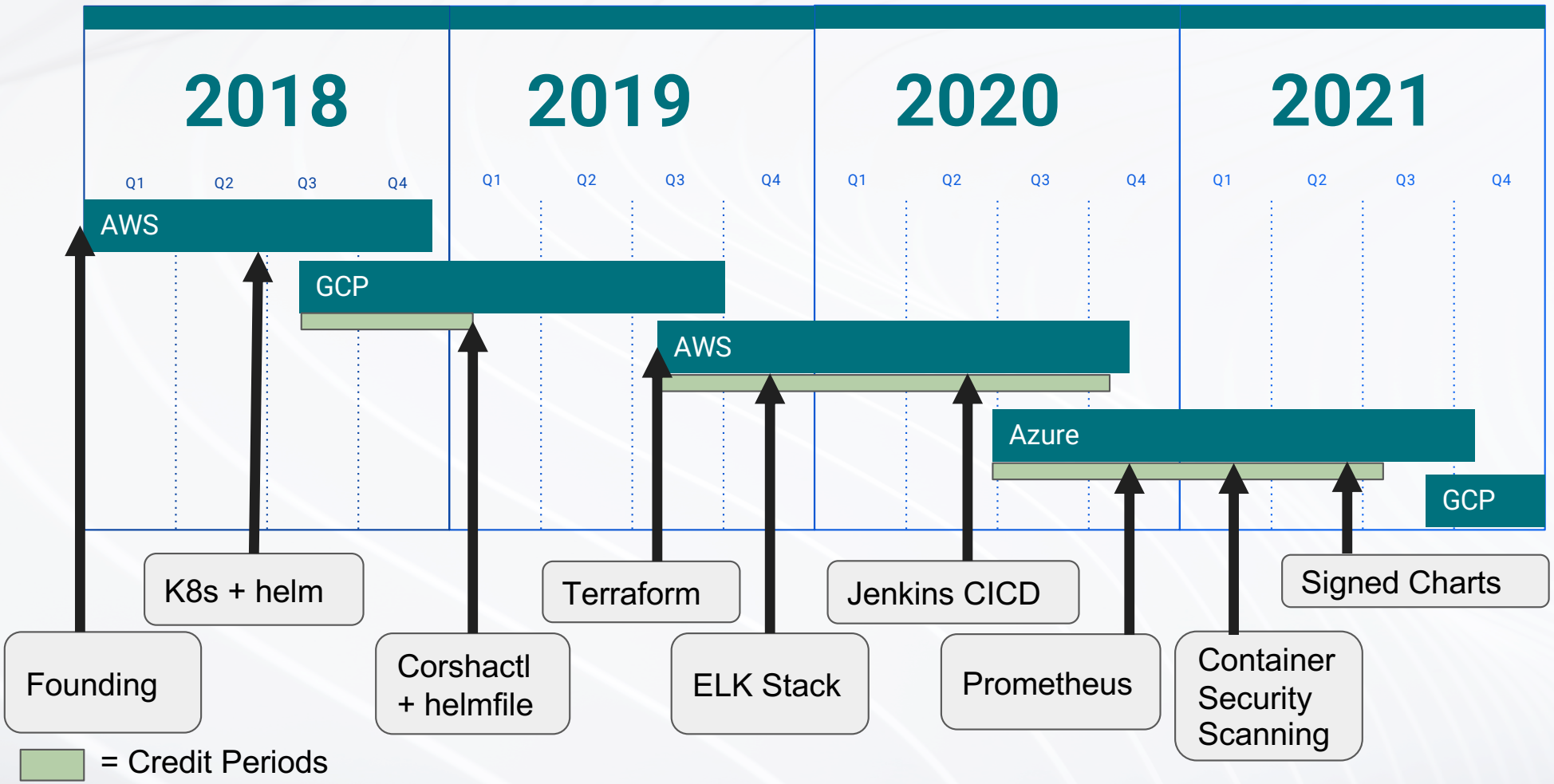
- Application architecture helps us with migrations
  - Blockchain nodes straddle cloud providers
  - Other examples
    - PgPool-II ‘switch over’ via pcp_attach_node and pcp_detach_node
- Disaster Recovery (DR)
  - COTS solutions like velero and/or k8s volume snapshots
  - Develop automation around creation and use of snapshots



Putting It All Together



Our Journey So Far



Lessons Learned

- Choose a persistence layer that has good distributed systems properties
  - Eases migration
  - Allows for ‘straddling’ cloud providers
- Migrate your dev cluster to the new cloud provider first
- Examine your cloud costs, don’t bother migrating low cost services, i.e. DNS
- Conduct extensive search for FOSS projects
- Concentrate provider specific config into a bootstrap Helm chart
- Pull as much off the shelf (esp. Terraform IAM stuff) as you can
- Requests/limits really help the scheduler

Not Just For Startups:
Scaling These Concepts

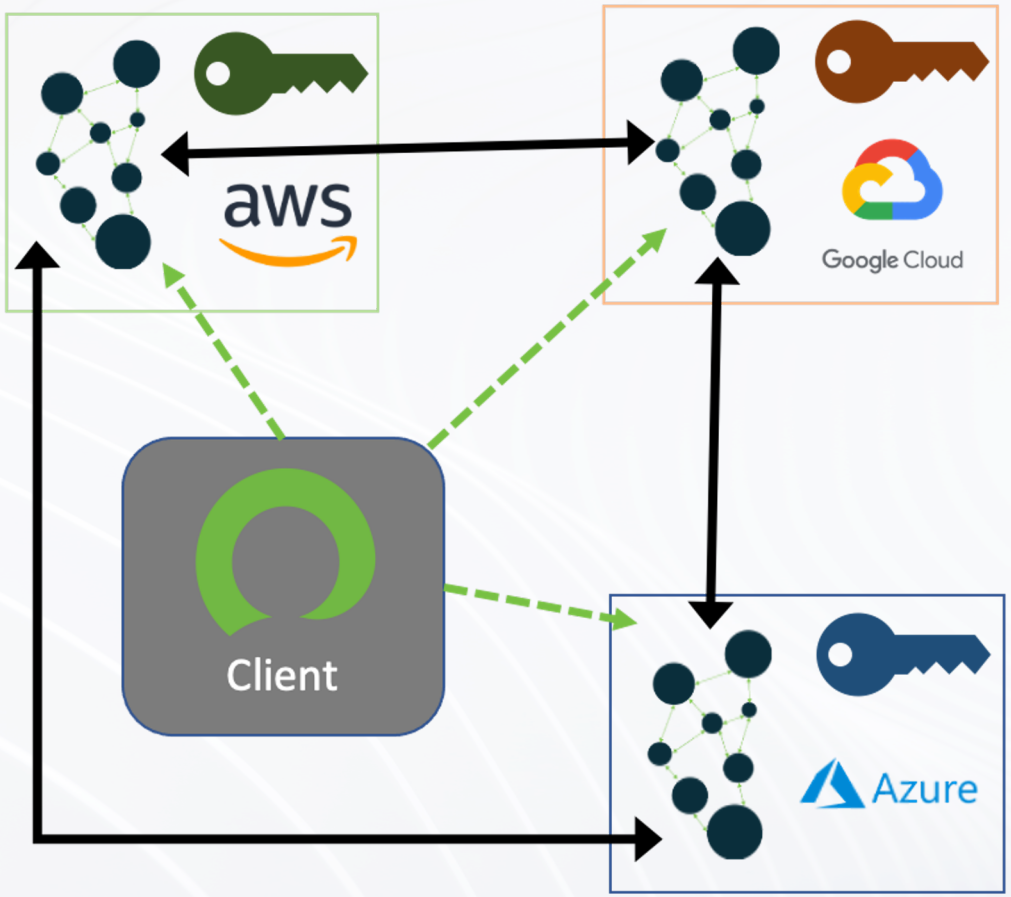
You can save > 20% by switching providers

Leverage your negotiating power

We are working with GCP on our next move now…

Looking Ahead: Multi-Cloud Deployments

- Provider journey driven by economics
- What if we went multi-cloud?
- Use cloud agnostic design to further business goals
- ‘Zero Trust Deployment’ to insure against reliance on single provider



Recap

- Significant savings if you comparison shop
- Migrating cloud providers doesn’t have to be painful
- Use cloud agnostic tooling
  - Minimize reliance on proprietary services
- Good distributed systems properties allowed us to ‘straddle’ cloud providers

Thank you!

Reach out to us to learn more!

Alex Meijer
alex@corsha.com

Anusha Iyer
anusha@corsha.com