AWS
re:Invent

CON332-S

Extreme infrastructure automation with Wavefront by VMware

Matthew Zeier
CTO
Spotinst

Kai Paro
Sr. DevOps Engineer
Wavefront by VMware

☒ The image part with relationship ID rId3 was not found in the file.

Continuous Cloud Optimization

spotinst

Speakers

Kai Paro

Sr. DevOps Engineer

WAVEFRONT by vmware

Kevin McGrath

CTO

spotinst

Company Snapshot

2015
Founded

intel

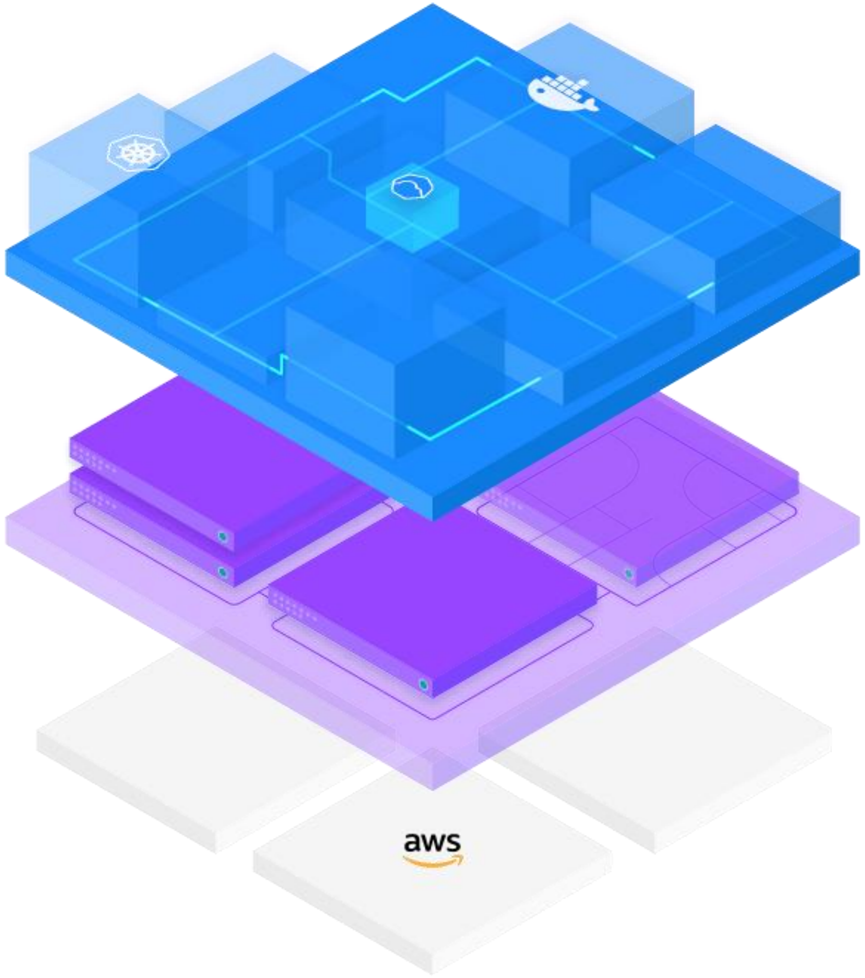
2017
Raised $17M

HIGHLAND
CAPITAL PARTNERS

2018
Raised $35M

Today
170+ Employees | 4 Offices
1,000+ Customers Worldwide

Spotinst **automates** cloud workloads to improve **performance**, reduce **complexity**, and lower compute infrastructure **costs** by 90%



One Stop Spot for Cloud Optimization

cloud analyzer

Cloud Management and Continuous Optimization
Where Finance and IT succeed.

eco

Continuous Cloud Commitment Management
Intelligent reserved and savings plans lifecycle automation with 75% cost optimization.

ocean

Serverless Containers
Your containers and zero infrastructure management with 90% cost optimization.

elastigroup

Cloud IaaS Optimization
Automate any application workload with 90% cost optimization.

managed instance

Optimized Pricing for Stateful, Single-Instance Workloads
Guaranteed data and IP persistence for your instance with 90% cost optimization.

tm

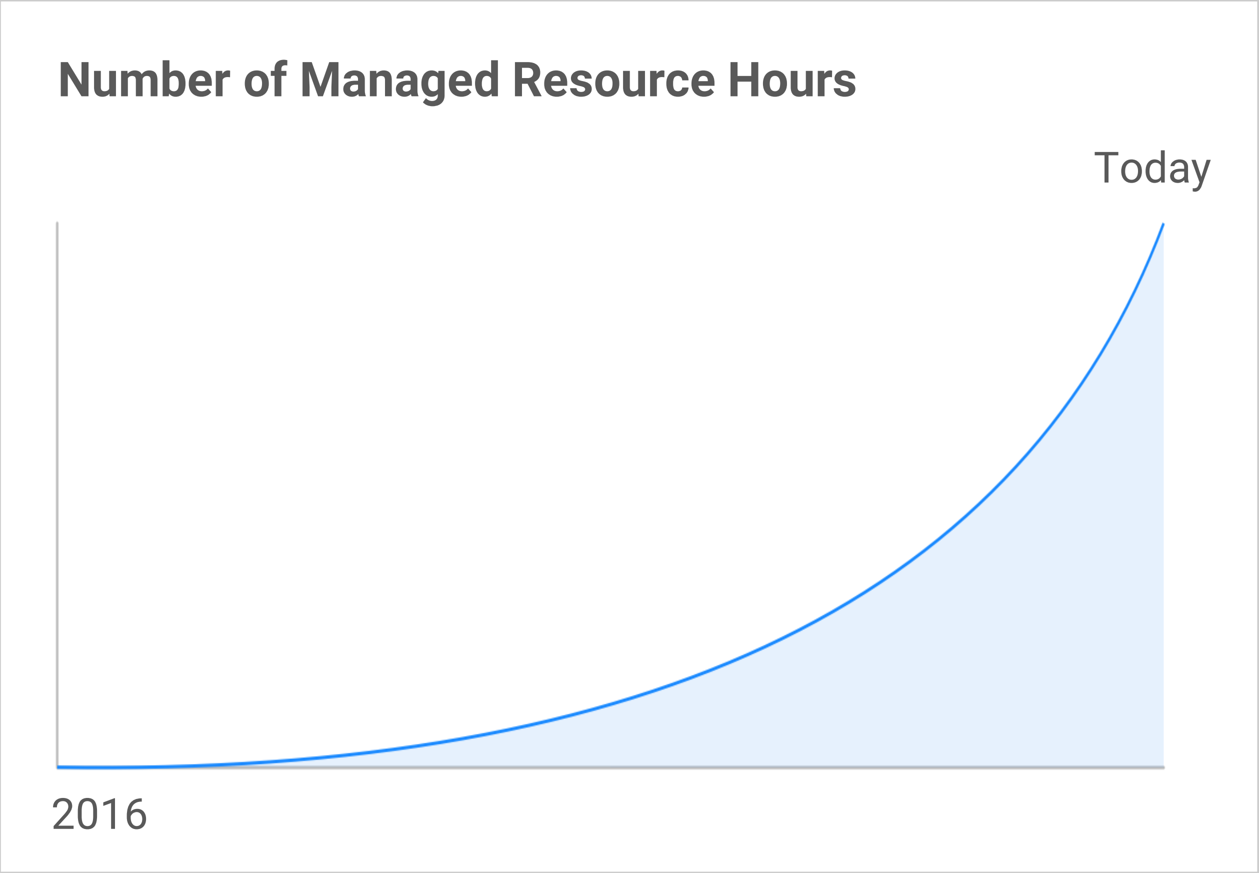
Strength in Numbers

3B+

Cloud Resource Hours/Month

Providing 60%–90% cost reduction

Hundreds of millions of dollars saved yearly



Deployed Worldwide in 50+ Countries Serving Enterprises and SMBs

fiverr®

SONY

Chegg®

SAMSUNG

ticketmaster®

WiX

cādence

DEMANDBASE

Hewlett Packard Enterprise

IBM

Unilever

vmware

AMERICAN FAMILY INSURANCE®

intel®

duolingo

UNIVERSITY OF NOTRE DAME

Spotinst Elastigroup | Cloud IaaS Optimization

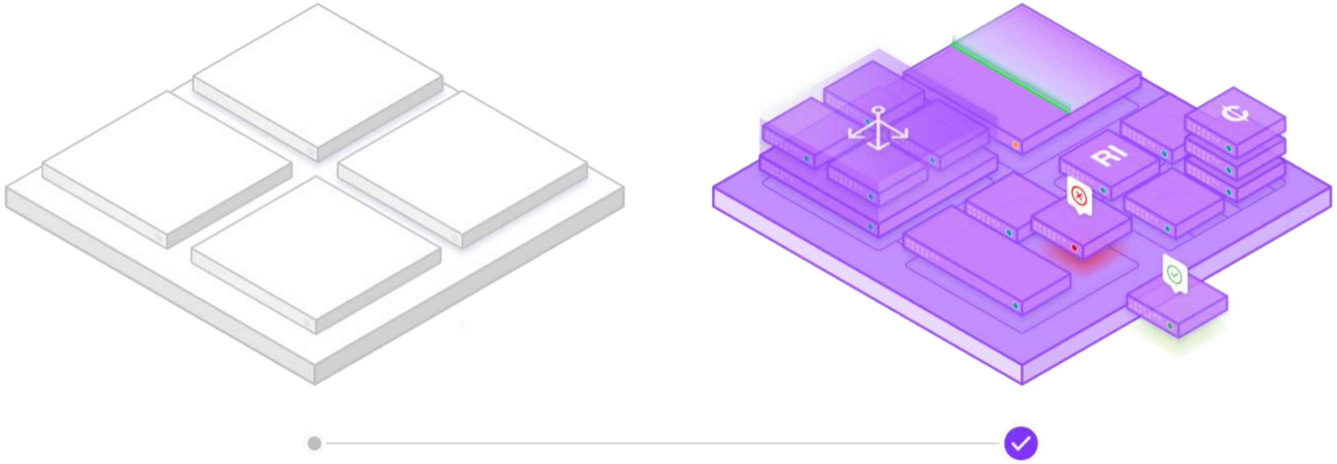
Automate any application workload with up to **90%** cost optimization.

Optimize Costs

Reliably leverage cloud excess capacity to optimize cost and save up to 90% on compute infrastructure.

Simplify Operations

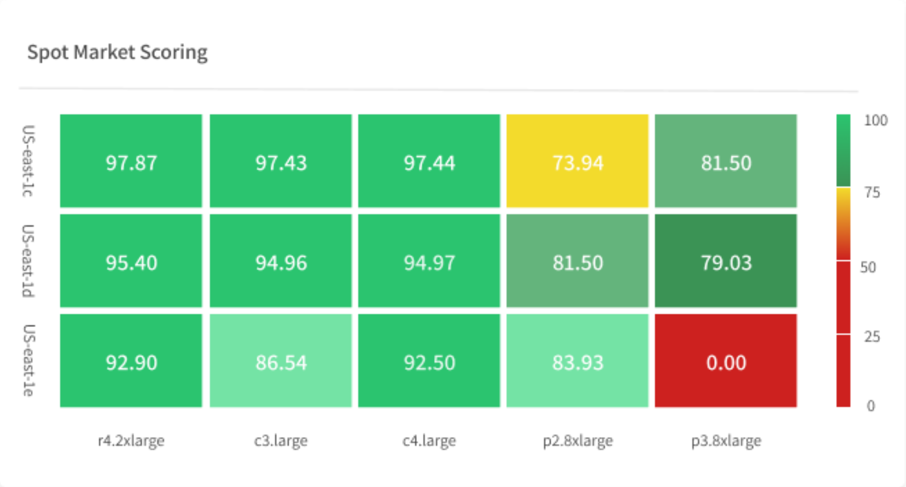
Scale, manage, and accelerate workloads without the complexity and risk of manually managing your infrastructure.



Spotinst Elastigroup

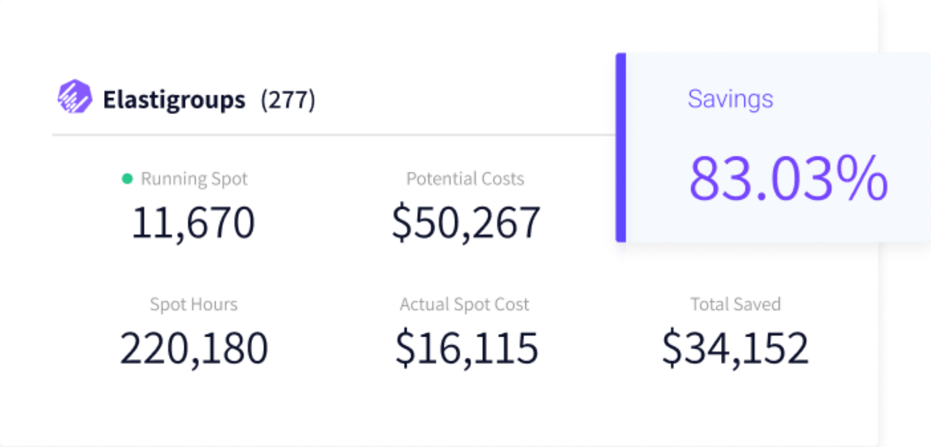
Prediction Is the Key

Elastigroup predicts Spot Instances behavior, capacity trends, pricing, and interruptions rate.



Spot Market Scoring

	r4.2xlarge	c3.large	c4.large	p2.8xlarge	p3.8xlarge
US-east-1c	97.87	97.43	97.44	73.94	81.50
US-east-1d	95.40	94.96	94.97	81.50	79.03
US-east-1e	92.90	86.54	92.50	83.93	0.00



Elastigroups (277)

Savings
83.03%

● Running Spot
11,670

Potential Costs
$50,267

Spot Hours
220,180

Actual Spot Cost
$16,115

Total Saved
$34,152

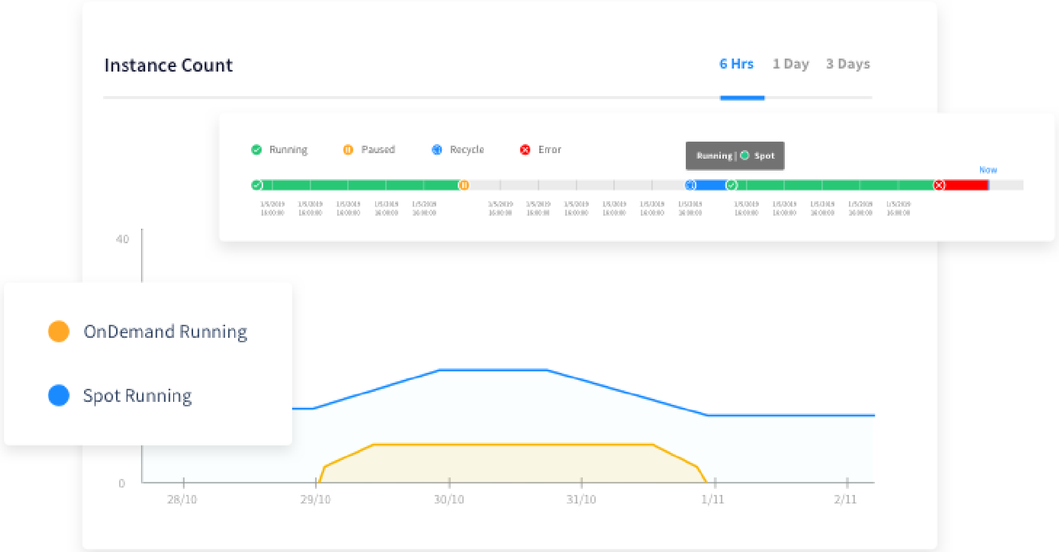
Up to 90% Cost Optimization with SLA

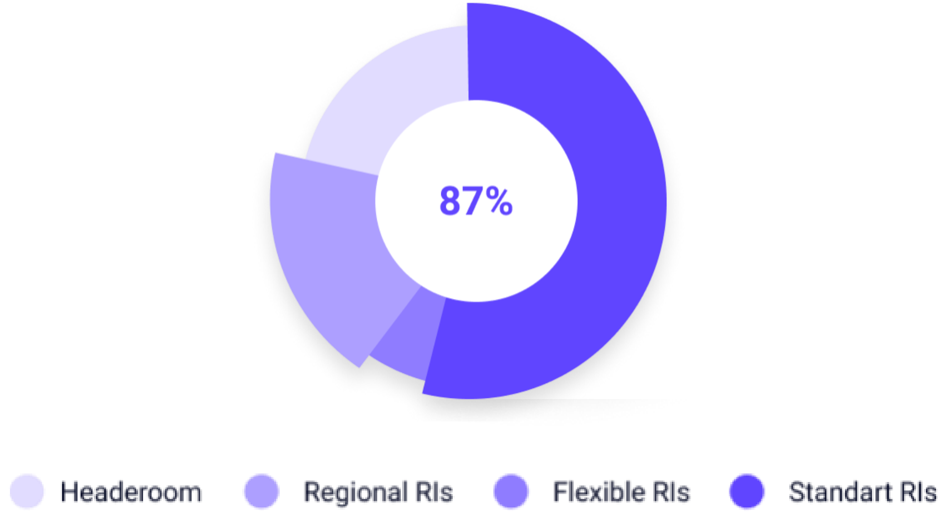
By predicting interruptions and fluctuations, Elastigroup is able to offensively rebalance clusters to prevent interruption.

Spotinst Elastigroup

Utilize Reserved Instances

Prioritize all underutilized reservations across your accounts and apply reservations discount usage prior to launching Spot or On-Demand Instances.



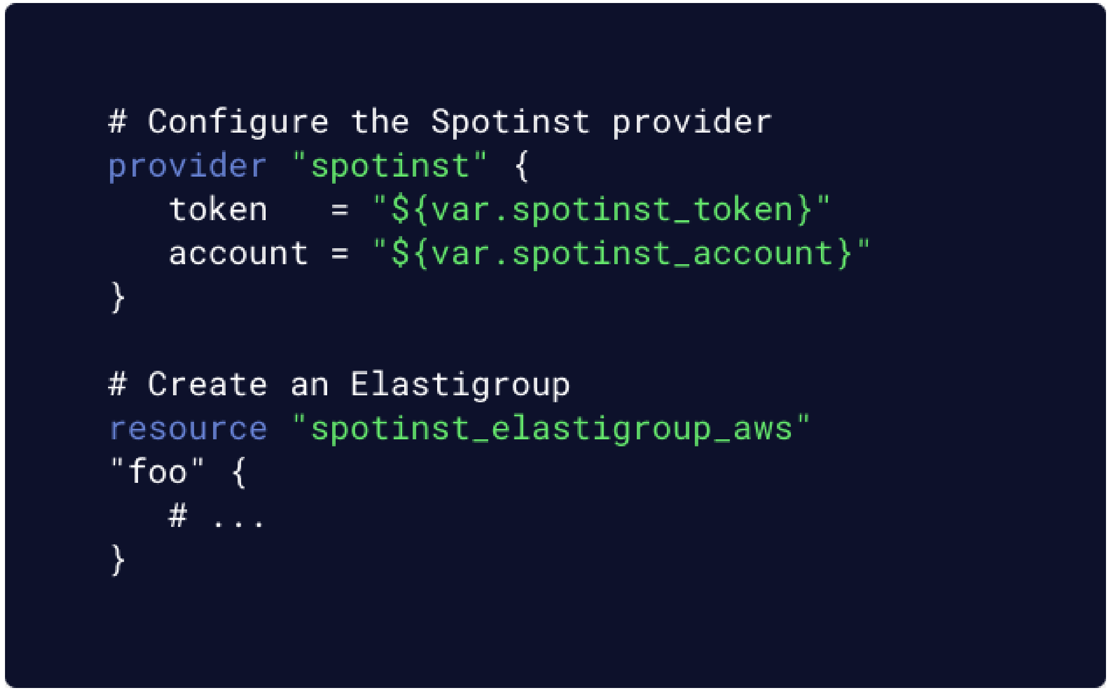


Enterprise-Grade SLA

In the event that Spot Instances aren't available, Elastigroup will automatically fall back to On-Demand Instances and will revert back to Spot Instances whenever possible, all while persisting your storage, network configuration, and state.

Connects with Your DevOps Tools & Stack

Seamlessly integrates with your existing IaaC (**Infrastructure-as-a-Code**) tools such as Ansible, Terraform, and AWS CloudFormation, so you will be able to apply an end-to-end automated process of your stack.



```
# Configure the Spotinst provider
provider "spotinst" {
   token   = "${var.spotinst_token}"
   account = "${var.spotinst_account}"
}

# Create an Elastigroup
resource "spotinst_elastigroup_aws"
"foo" {
   # ...
}
```

Spotinst Eco | Continuous Reserved Capacity Management

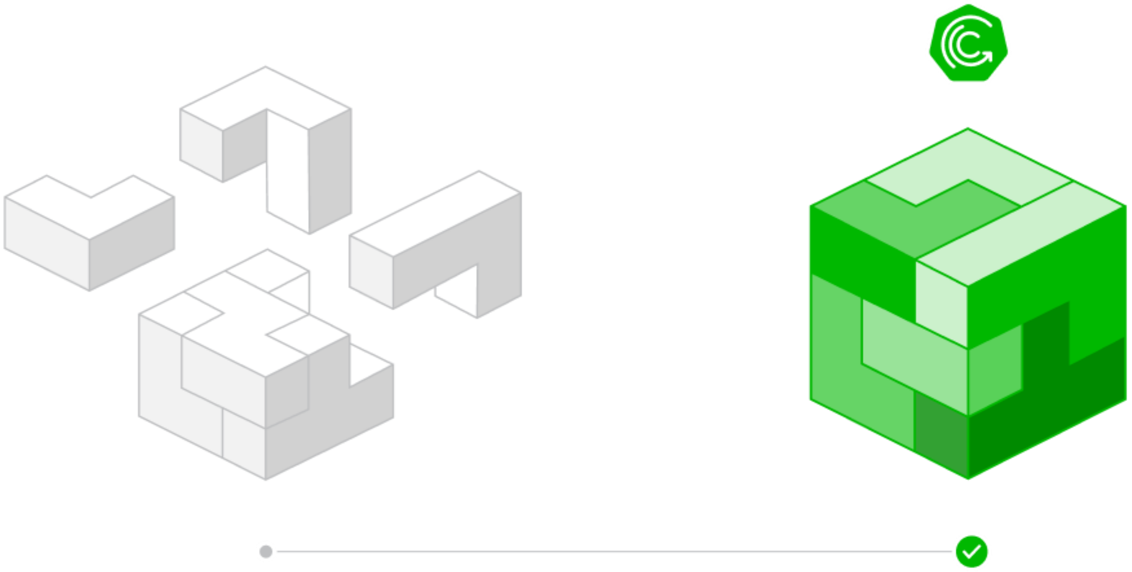
Intelligent RI and savings plans lifecycle automation with 75% cost optimization.

Managed RI Lifecycle

Comprehensive analysis of compute workloads; RI buying and selling in AWS Marketplace is automated to ensure that your workload is running at optimal pricing.

Finance & DevOps Synergy

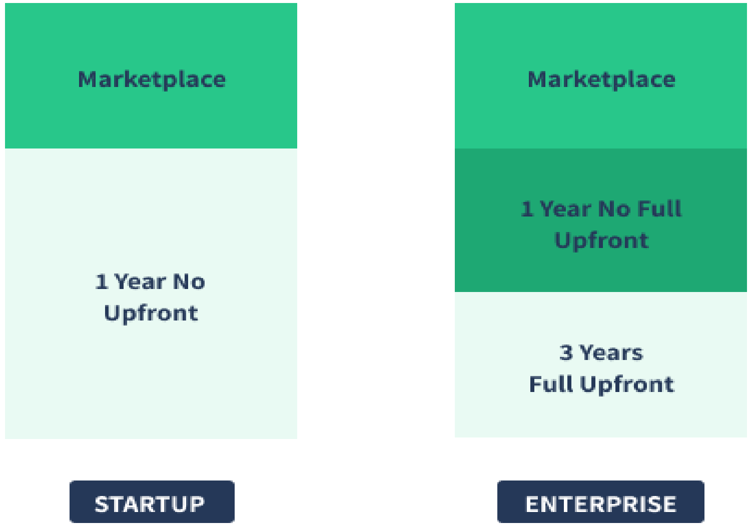
With full visibility into compute consumption and automation of optimal RI and savings plans strategies, finance and DevOps teams can easily collaborate on managing cloud cost.



Spotinst Eco

No Engineering Effort

Reserved instances (RIs) are a billing construct. Engineers don’t have to change anything about the compute or applications they use today. Once enabled, Spotinst Eco will continually track usage as well as build forecasting models to constantly manage the lifecycle.



Diversify Commitment

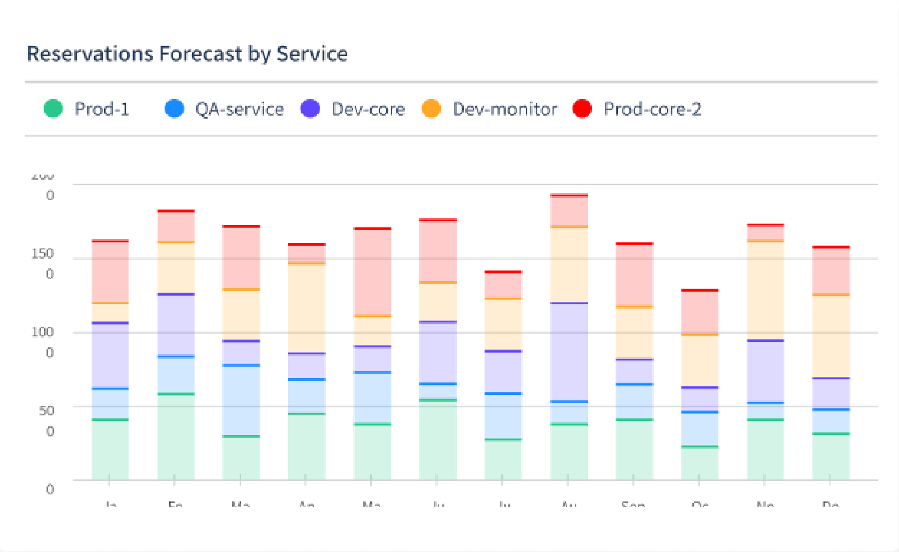
Eco acts as an RIs broker, utilizing the Marketplace and mixing and matching commitment lengths from 2 to 36 months so that utilization will be optimal with as little commitment as possible.

Spotinst Eco

Spotinst RI Marketplace

With hundreds of accounts and hundreds of thousands of RIs under management, Spotinst Eco can quickly match customers who have immediate needs to buy and sell reservations on the Marketplace, acting as a perfect RI broker.

Reserved Instances Availability
100
90
80
70
60
50
40
30
20
10
0



Forecast Powered by Machine Learning

As smart as a human can be, forecasting cloud commitment in an increasingly complicated cloud environment is inefficient, even when using the best reporting tools out there. Eco continually analyzes millions of data points to identify the makeup of your ideal RI fleet.

Spotinst Managed Instances

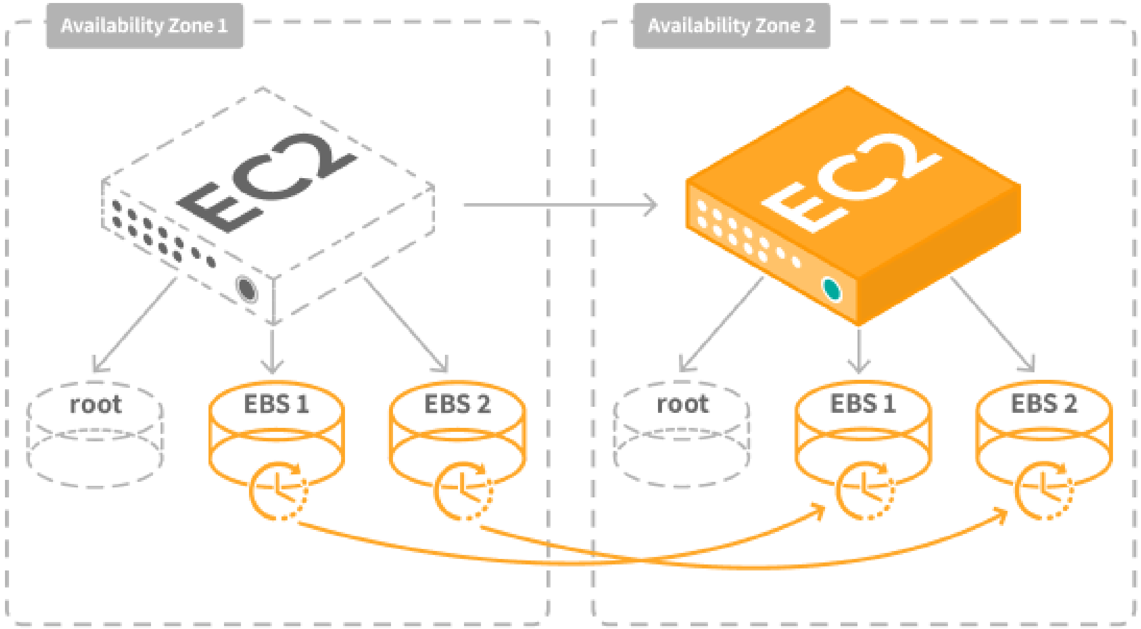
Intelligent Resource Persistent of

Root Volume

Storage Volumes

Private IP

Public IP



Spotinst Ocean | Serverless Containers

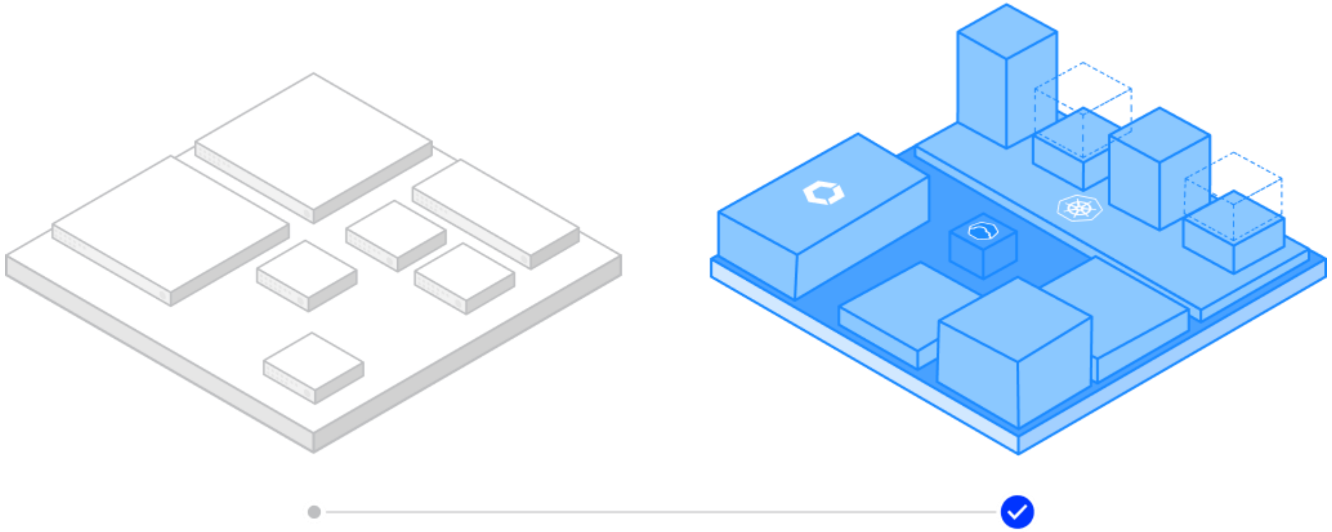
Deploy containers on abstracted infrastructure with up to **90%** cost optimization.

Container-Driven Autoscaling

Auto-detect pod or task infrastructure requirements so the appropriate instance size or type will always be available.

Simplify Operations

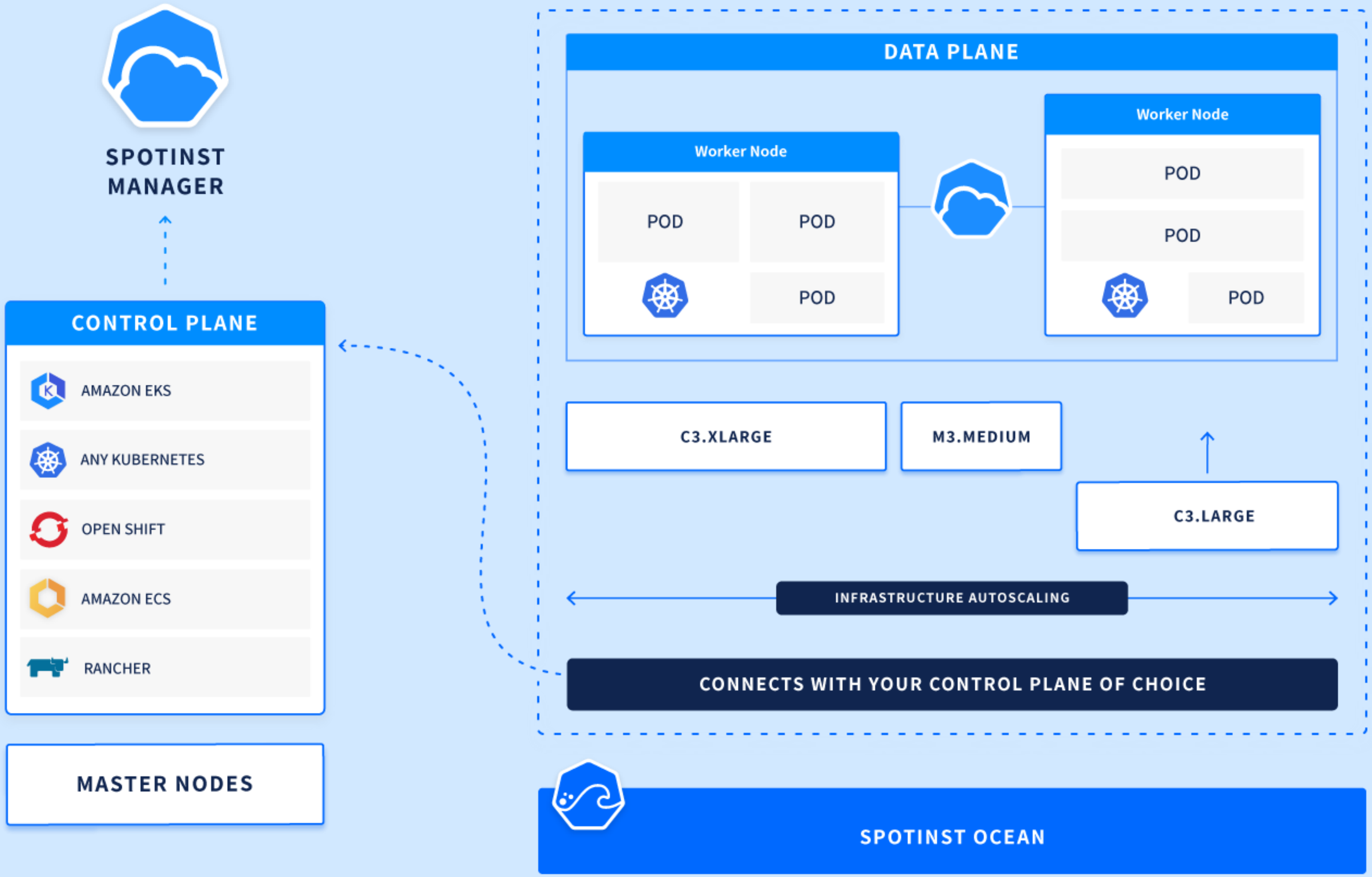
Deploy more without having to manage all the details of the underlying container infrastructure.



Why Container?

- Platform independence: Build it once, run it anywhere
    - Abstraction of OS and underlying infrastructure
- Lightweight and efficient
- VMs can be gigabytes while containers can be mere megabytes
- Easy to package and deploy
- Effective isolation and resource sharing
- Speed: Start, create, replicate, or destroy containers in seconds
- Improved developer productivity and development pipeline

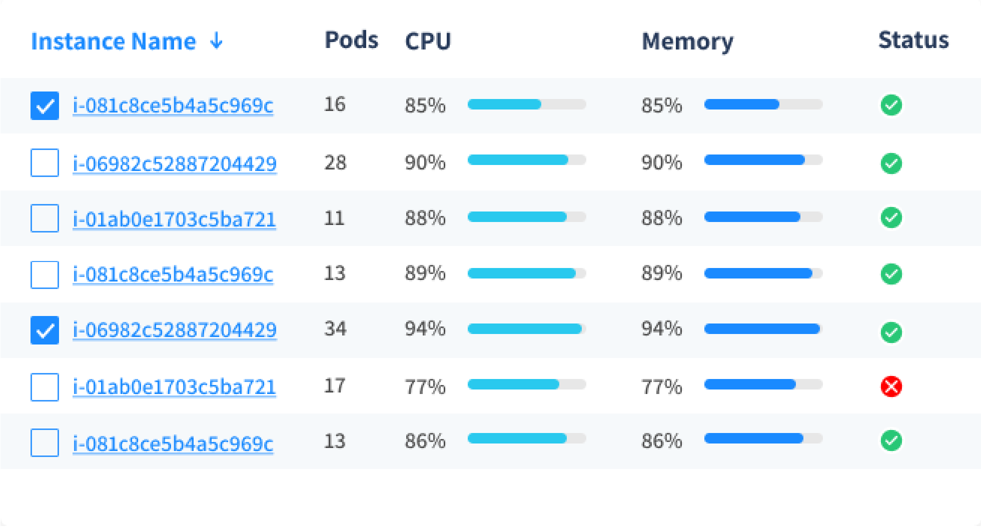
Kubernetes Data Plane Management



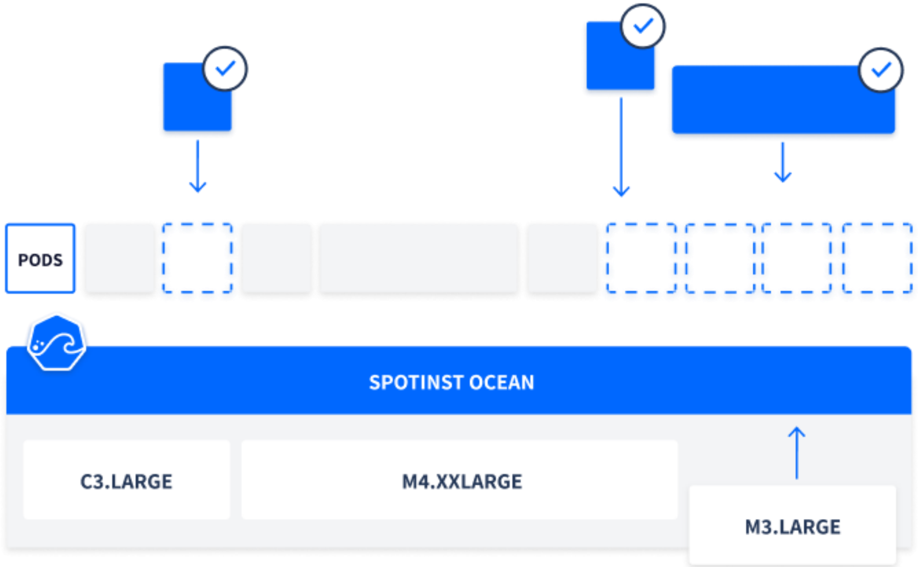
Spotinst Ocean

Container-Driven Infrastructure Autoscaling

Ensures that all pods/tasks have resources to run on, and systematically selects the most suitable instance type that will facilitate the containers' requirements.



	Instance Name ↓	Pods	CPU	Memory	Status
☑	i-081c8ce5b4a5c969c	16	85%	85%	✓
□	i-06982c52887204429	28	90%	90%	✓
□	i-01ab0e1703c5ba721	11	88%	88%	✓
□	i-081c8ce5b4a5c969c	13	89%	89%	✓
☑	i-06982c52887204429	34	94%	94%	✓
□	i-01ab0e1703c5ba721	17	77%	77%	✗
□	i-081c8ce5b4a5c969c	13	86%	86%	✓



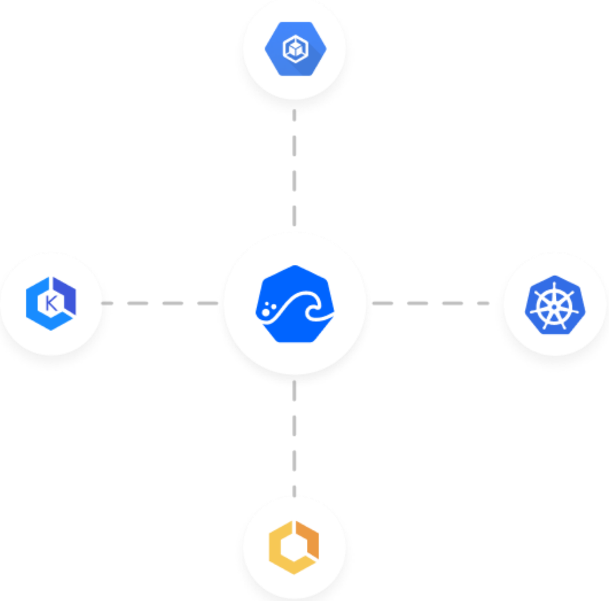
Maximize Resource Utilization

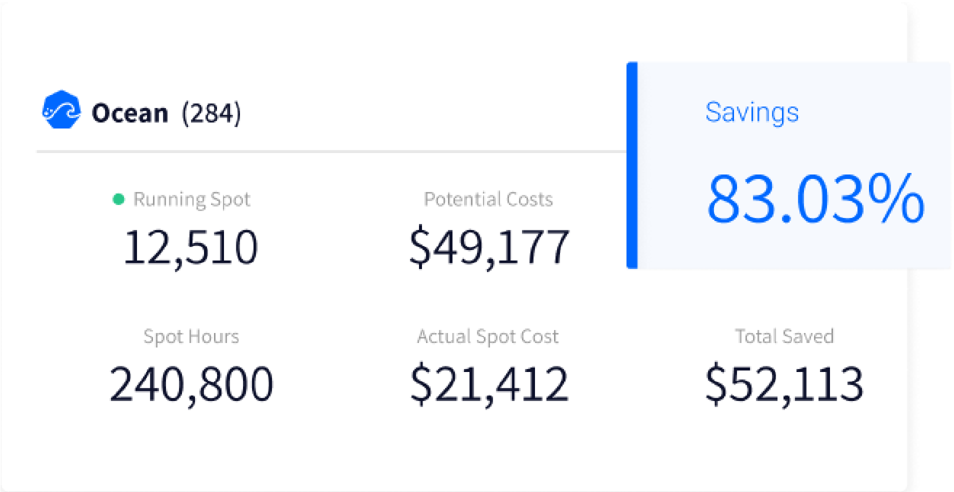
Validates that your instances are fully utilized before spinning up new ones, thus enabling an additional layer of cost efficiency.

Spotinst Ocean

Bring Your Own Control Plane

Ocean seamlessly integrates and supports your stack, whether you are using Amazon ECS or Kubernetes orchestrators such as Amazon EKS.





Ocean (284)

Running Spot
12,510

Potential Costs
$49,177

Savings
83.03%

Spot Hours
240,800

Actual Spot Cost
$21,412

Total Saved
$52,113

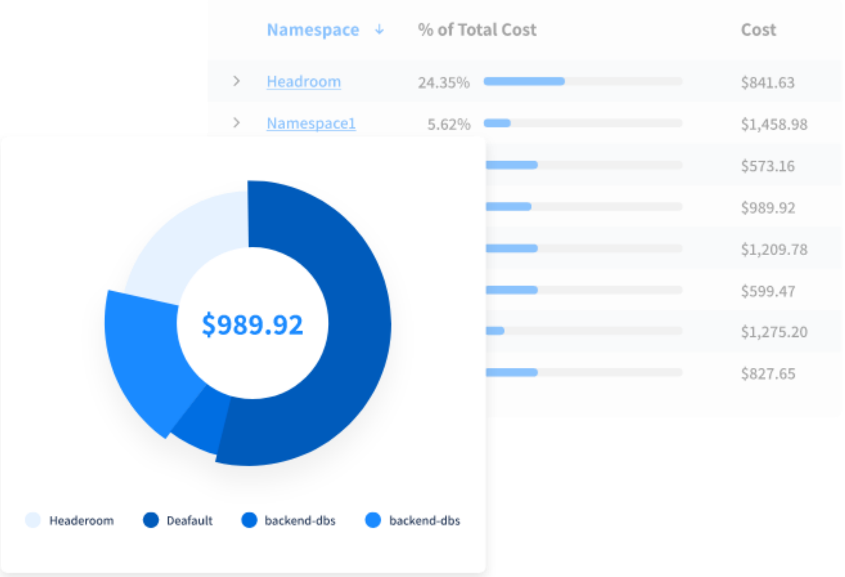
Save up to 90% on Infrastructure Costs

Optimize up to 90% of your cloud compute costs by leveraging cloud excess capacity as the underlying infrastructure that facilitates your containers allocation, while enabling the option to fall back to On-Demand.

Spotinst Ocean

Cost Showback

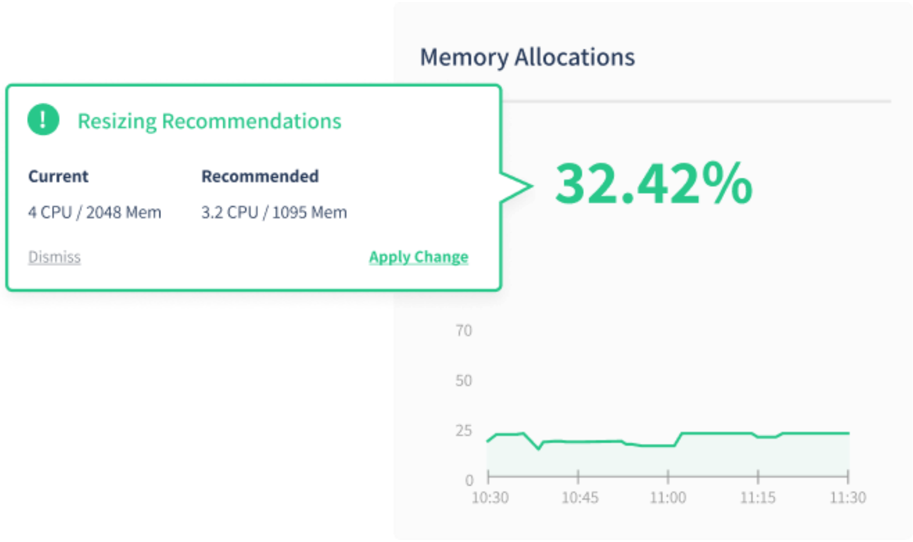
Get a more granular view of the cluster’s cost breakdown (compute and storage) for each and every one of the cluster’s resources, such as deployment/service, cron jobs, tasks, and pods.



Namespace ↓	% of Total Cost	Cost
Headroom	24.35%	$841.63
Namespace1	5.62%	$1,458.98
		$573.16
		$989.92
		$1,209.78
		$599.47
		$1,275.20
		$827.65

Vertical Container Autoscaling

Measuring in real time the CPU/memory of pods provides resource-actionable suggestions based on the consumption in your cluster.



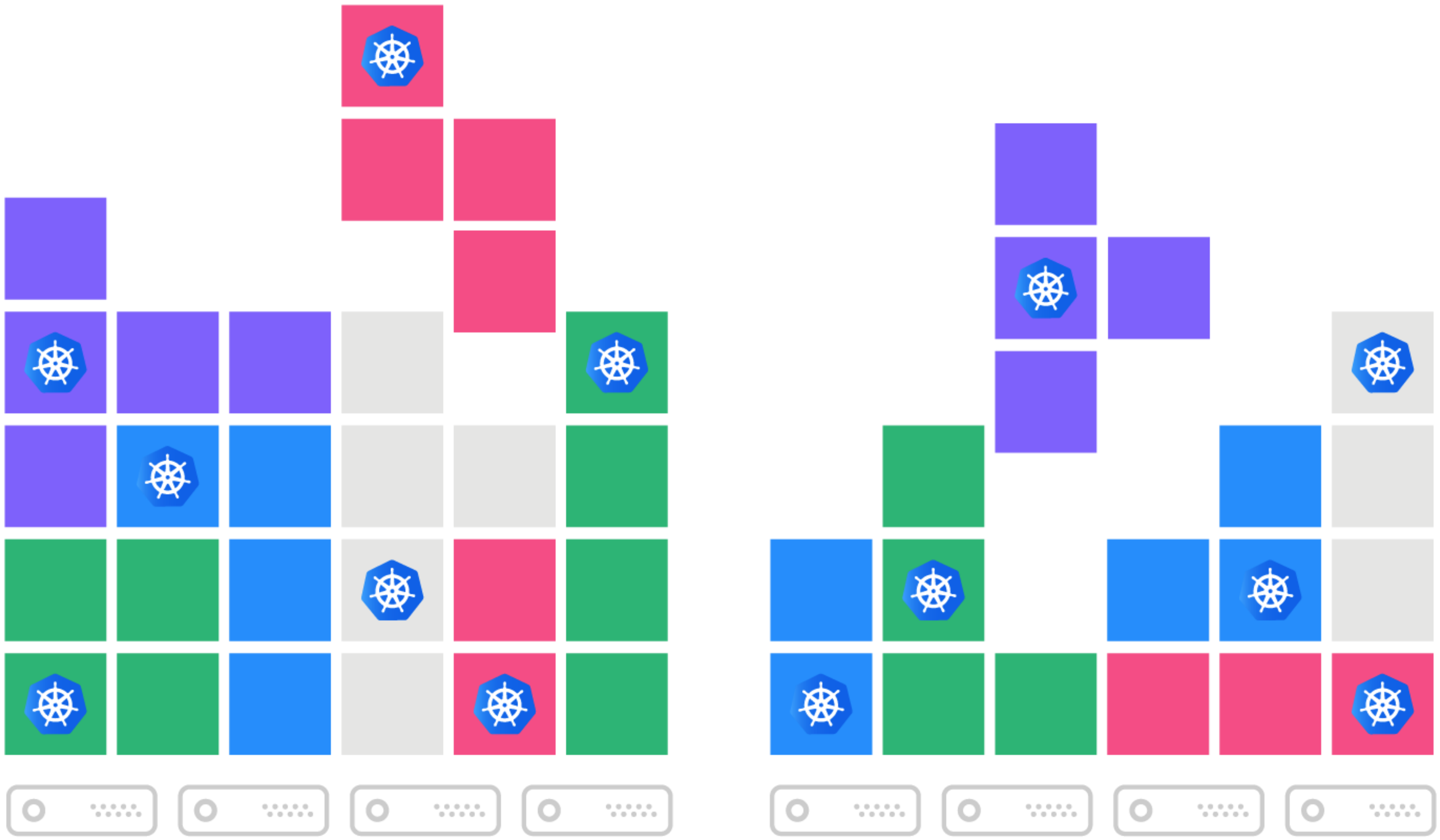
Memory Allocations

Resizing Recommendations

Current	Recommended
4 CPU / 2048 Mem	3.2 CPU / 1095 Mem

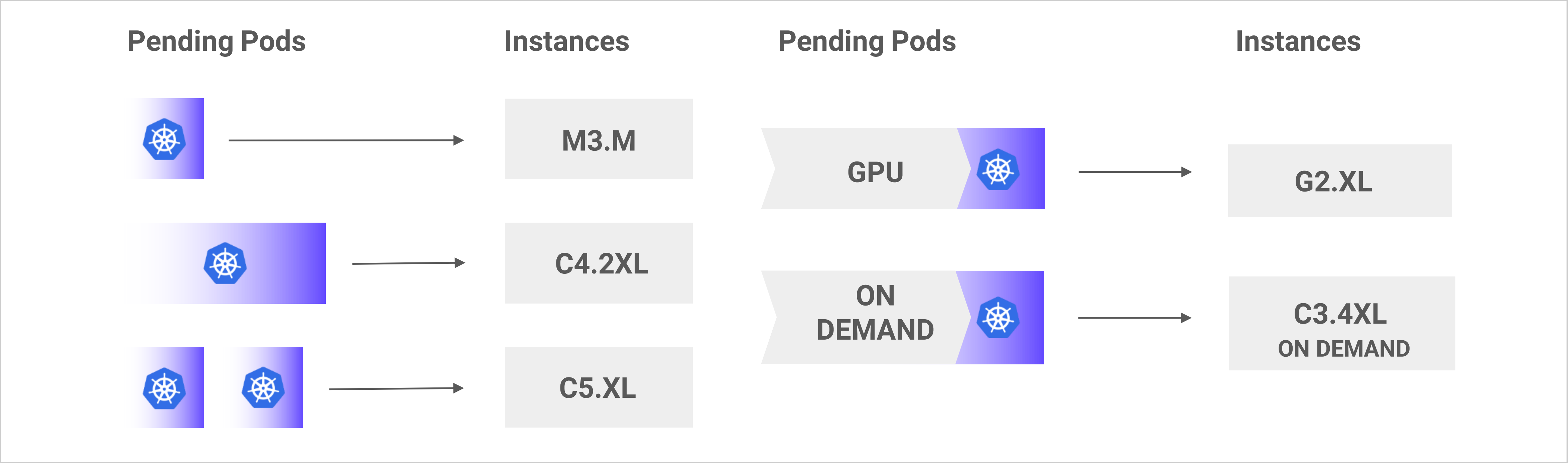
Dismiss   Apply Change

Kubernetes | Tetris Scaling



Containers Are First-Class Citizens

Instance size, type, and lifecycle are determined based on the pod/task requirements while honoring labels, taints, and tolerations.



Simplicity and Automation for Enterprise Workloads

- No VMs to manage
- No need to choose instance types/sizes
- 80% less on infrastructure costs by reliably leveraging spot/pre-emptible VMs
- Robust UI and API for Kubernetes monitoring and management
- Infrastructure autoscaling based on actual containers consumption

Patent pending

> Spotinst enables customers to move additional workloads to Spot Instances with less effort and greater confidence.

Joshua Burgin | General Manager, Amazon EC2

aws

Spotinst | Putting It All Together

- Three-layer approach to optimizing and automating container workloads
- Pricing model
- Spot, On-Demand, and Reserved Instances
- Instance sizing
- Matching pods to instances
- Container utilization
- Monitoring real usage

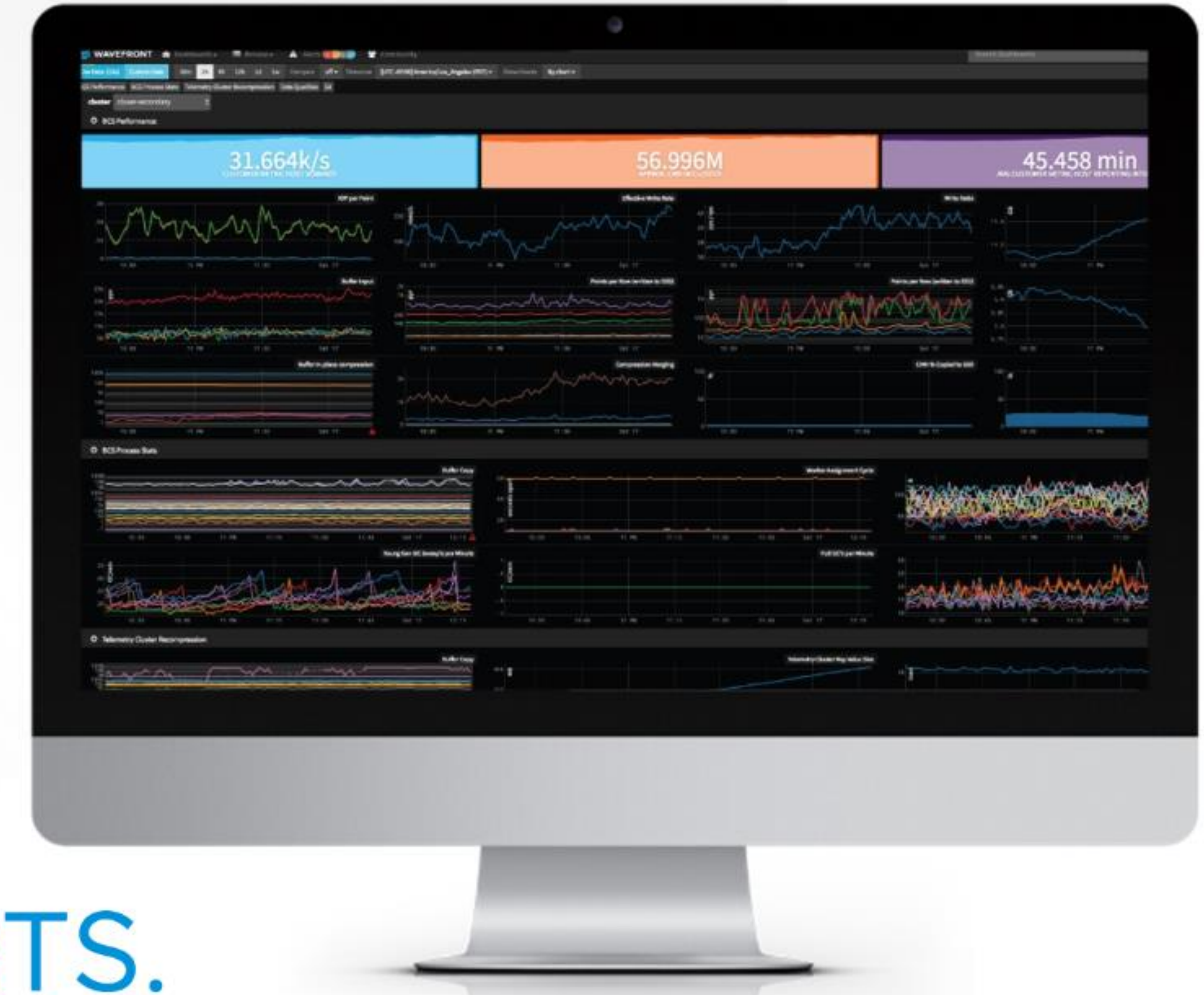
WAVEFRONT

by vmware

Overview

WAVEFRONT

by vmware



SPIFFY CHARTS.
INTELLIGENT ALERTS.

The Journey

2013

Wavefront Founded

2015

Exited

2017

VMware Acquires Wavefront

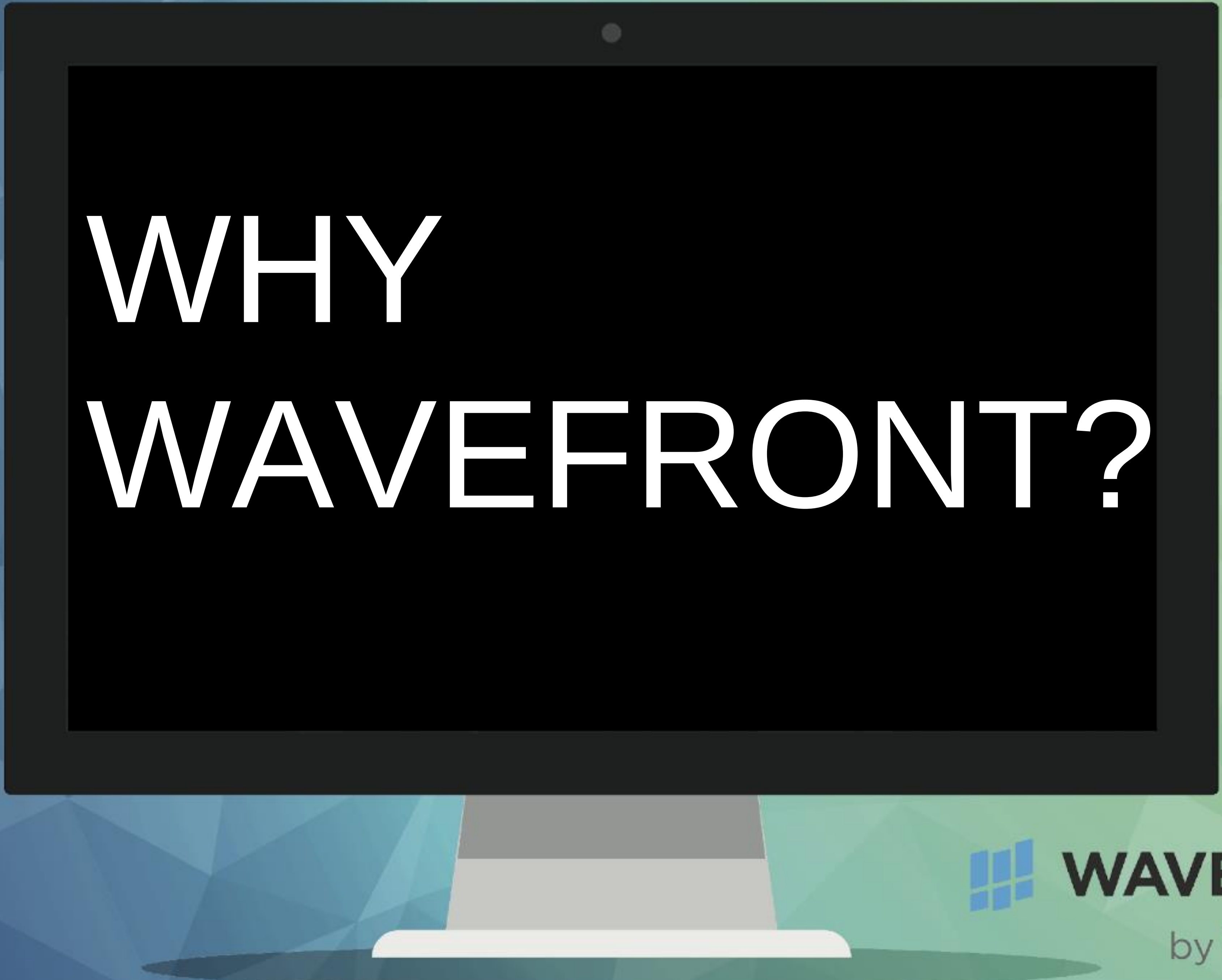
2018

Public Launch of 3D Observability (Metrics, Histogram, Tracing) and AI Genie

The Wavefront Effect

1	Unified Full Stack View
30+%	Reduction in Tooling Complexity
10x	Earlier Issue Detection
5x	Lower Prices than Traditional APM
100B+	Data Points Ingested Per Day (at Scale)

WHY WAVEFRONT?



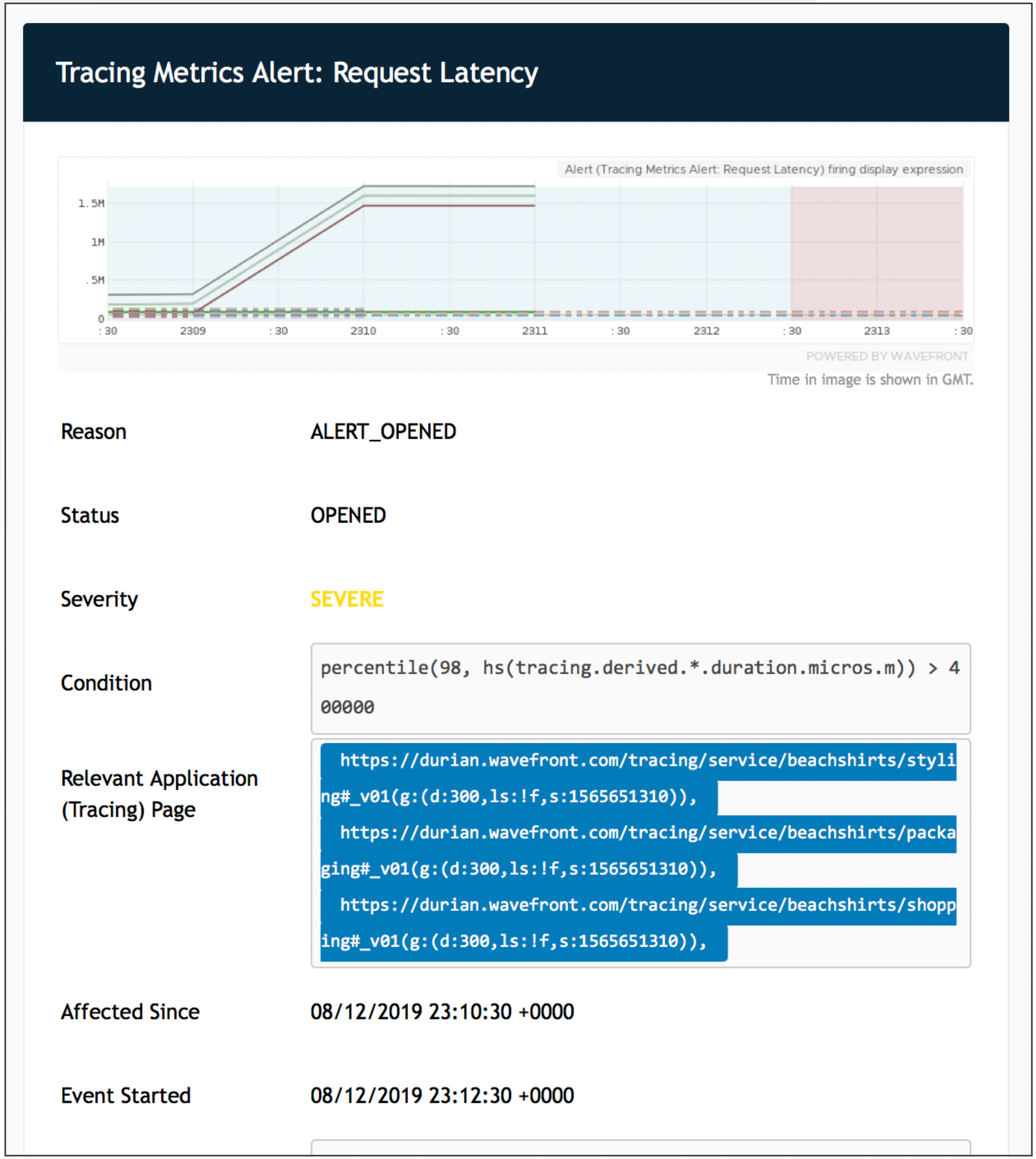
WAVEFRONT

by vmware

MAKING SENSE OF DATA IS HARD.

THIS IS WHY WAVEFRONT

WAVEFRONT by vmware®



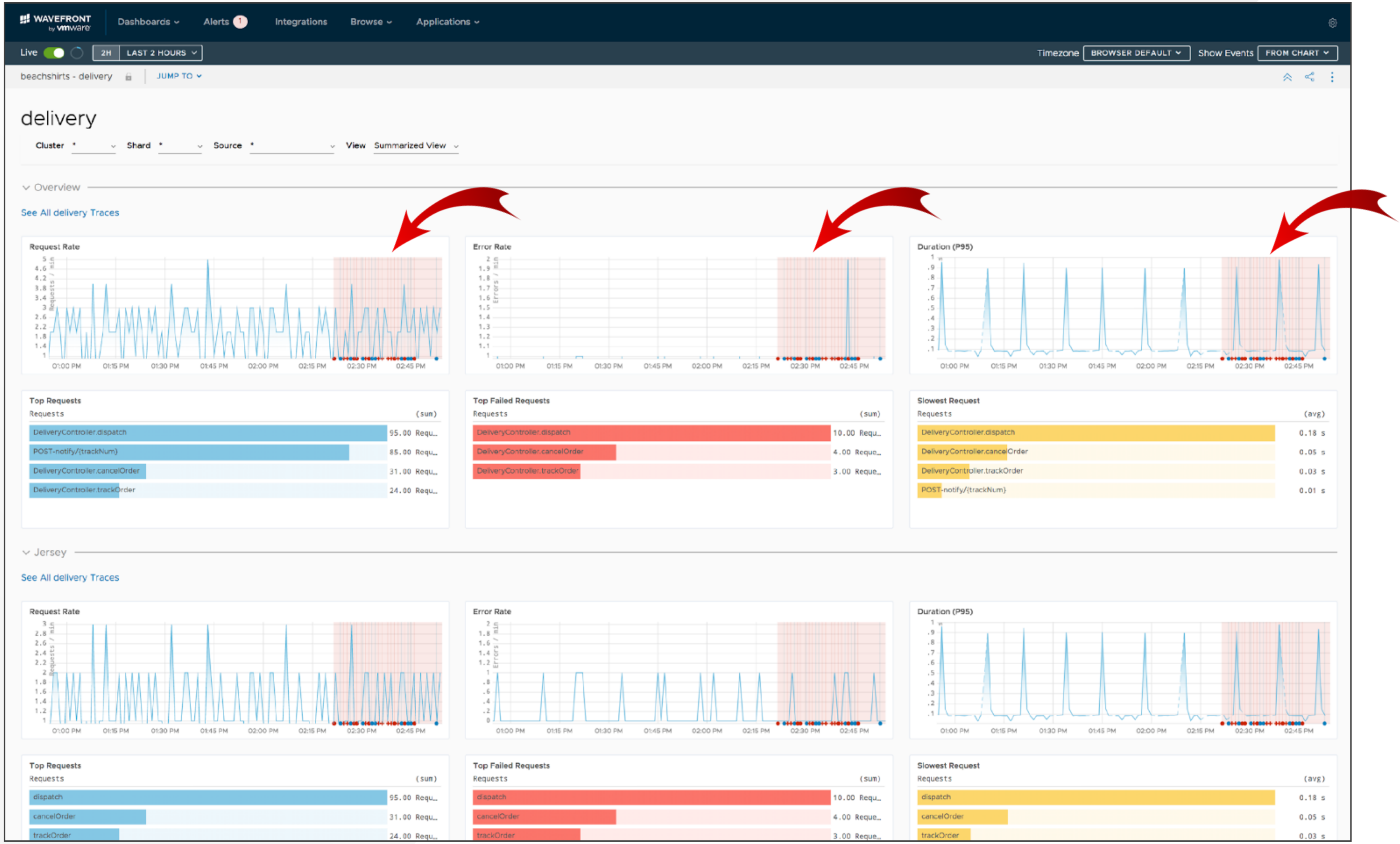
Alert (Tracing Metrics Alert: Request Latency) firing display expression
1.5M
1M
.5M
0
:30
2309
:30
2310
:30
2311
:30
2312
:30
2313
:30
POWERED BY WAVEFRONT

Tracing Metrics Alert: Request Latency

Time in image is shown in GMT.

Reason	ALERT_OPENED
Status	OPENED
Severity	SEVERE
Condition	`percentile(98, hs(tracing.derived.*.duration.micros.m)) > 400000`
Relevant Application (Tracing) Page	https://durian.wavefront.com/tracing/service/beachshirts/styling#_v01(g:(d:300,ls:!f,s:1565651310)), https://durian.wavefront.com/tracing/service/beachshirts/packaging#_v01(g:(d:300,ls:!f,s:1565651310)), https://durian.wavefront.com/tracing/service/beachshirts/shopping#_v01(g:(d:300,ls:!f,s:1565651310)),
Affected Since	08/12/2019 23:10:30 +0000
Event Started	08/12/2019 23:12:30 +0000

Context-Enriched Alerting Enables One-Click Troubleshooting



delivery

Cluster * Shard * Source * View Summarized View

Overview

See All delivery Traces

Top Requests

Requests	(sum)
DeliveryController.dispatch	95.00 Requ...
POST-notify/{trackNum}	85.00 Requ...
DeliveryController.cancelOrder	31.00 Requ...
DeliveryController.trackOrder	24.00 Requ...

Top Failed Requests

Requests	(sum)
DeliveryController.dispatch	10.00 Requ...
DeliveryController.cancelOrder	4.00 Reque...
DeliveryController.trackOrder	3.00 Reque...

Slowest Request

Requests	(avg)
DeliveryController.dispatch	0.18 s
DeliveryController.cancelOrder	0.05 s
DeliveryController.trackOrder	0.03 s
POST-notify/{trackNum}	0.01 s

Jersey

See All delivery Traces

Top Requests

Requests	(sum)
dispatch	95.00 Requ...
cancelOrder	31.00 Requ...
trackOrder	24.00 Requ...

Top Failed Requests

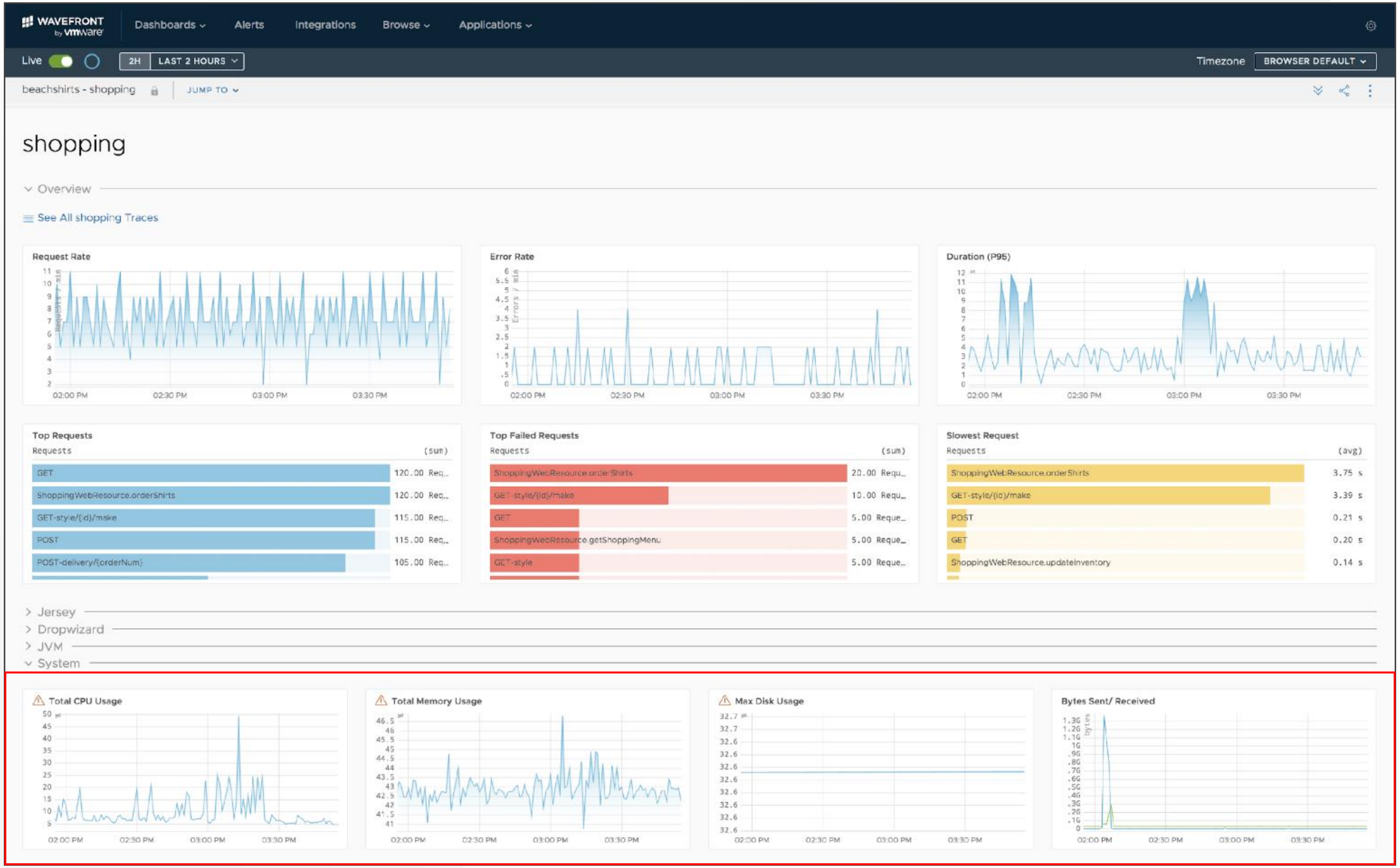
Requests	(sum)
dispatch	10.00 Requ...
cancelOrder	4.00 Reque...
trackOrder	3.00 Reque...

Slowest Request

Requests	(avg)
dispatch	0.18 s
cancelOrder	0.05 s
trackOrder	0.03 s

Troubleshoot Faster with Unified Views of Application & Infrastructure

Application Metrics



shopping

Overview

See All shopping Traces

Top Requests

Requests	(sum)
GET	120.00 Req…
ShoppingWebResource.orderShirts	120.00 Req…
GET-style/{id}/make	115.00 Req…
POST	115.00 Req…
POST-delivery/{orderNum}	105.00 Req…

Top Failed Requests

Requests	(sum)
ShoppingWebResource.orderShirts	20.00 Requ…
GET-style/{id}/make	10.00 Requ…
GET	5.00 Reque…
ShoppingWebResource.getShoppingMenu	5.00 Reque…
GET-style	5.00 Reque…

Slowest Request

Requests	(avg)
ShoppingWebResource.orderShirts	3.75 s
GET-style/{id}/make	3.39 s
POST	0.21 s
GET	0.20 s
ShoppingWebResource.updateInventory	0.14 s

Jersey

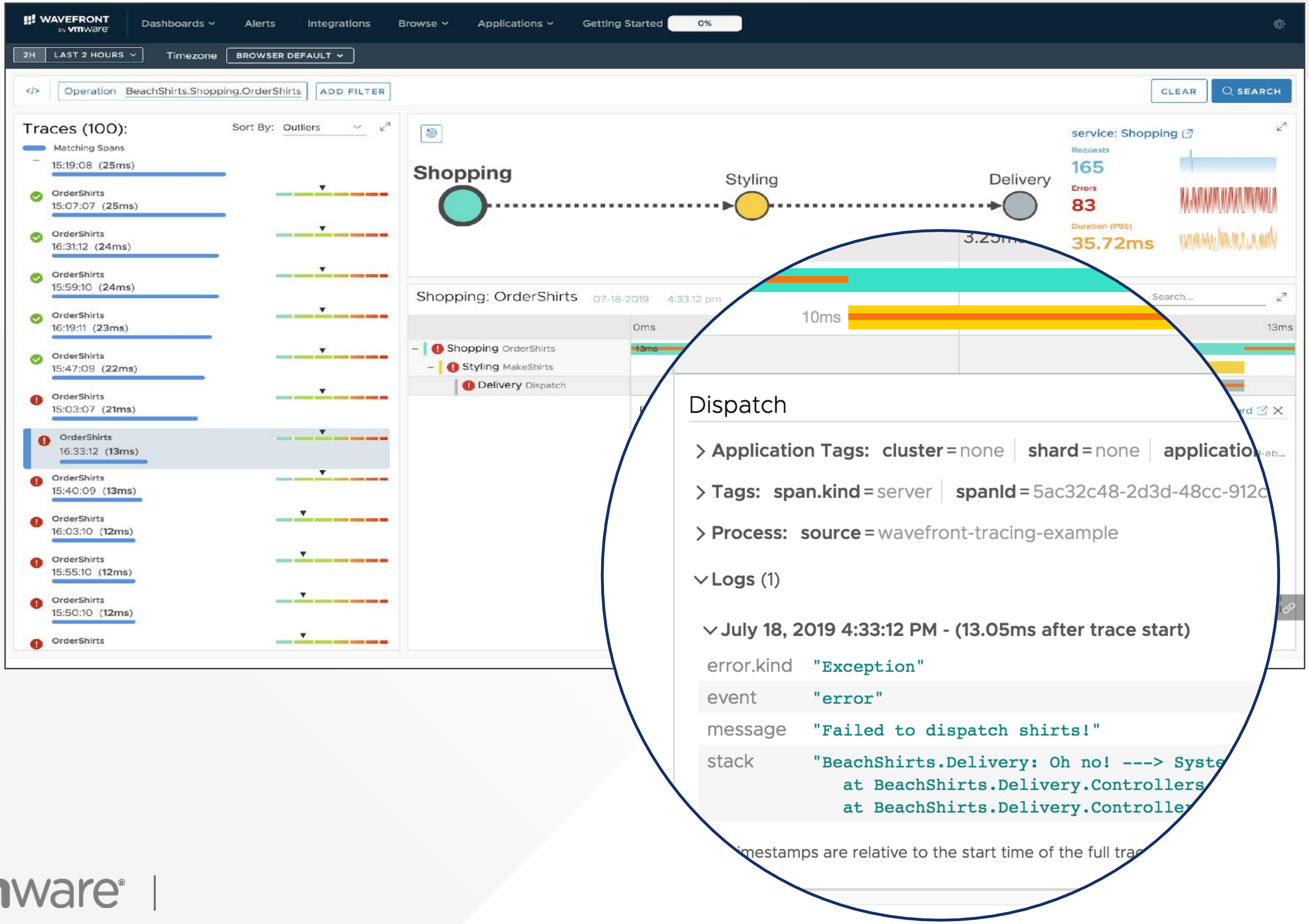
Dropwizard

JVM

System

System Metrics

Find Root Cause Faster with Span Logs



Traces (100):

	15:19:08 (25ms)
OrderShirts	15:07:07 (25ms)
OrderShirts	16:31:12 (24ms)
OrderShirts	15:59:10 (24ms)
OrderShirts	16:19:11 (23ms)
OrderShirts	15:47:09 (22ms)
OrderShirts	15:03:07 (21ms)
OrderShirts	16:33:12 (13ms)
OrderShirts	15:40:09 (13ms)
OrderShirts	16:03:10 (12ms)
OrderShirts	15:55:10 (12ms)
OrderShirts	15:50:10 (12ms)
OrderShirts	

Shopping: OrderShirts 07-18-2019 4:33:12 pm

- Shopping OrderShirts
  - Styling MakeShirts
    - Delivery Dispatch

Dispatch

> Application Tags: cluster = none | shard = none | applicatio...

> Tags: span.kind = server | spanId = 5ac32c48-2d3d-48cc-912c

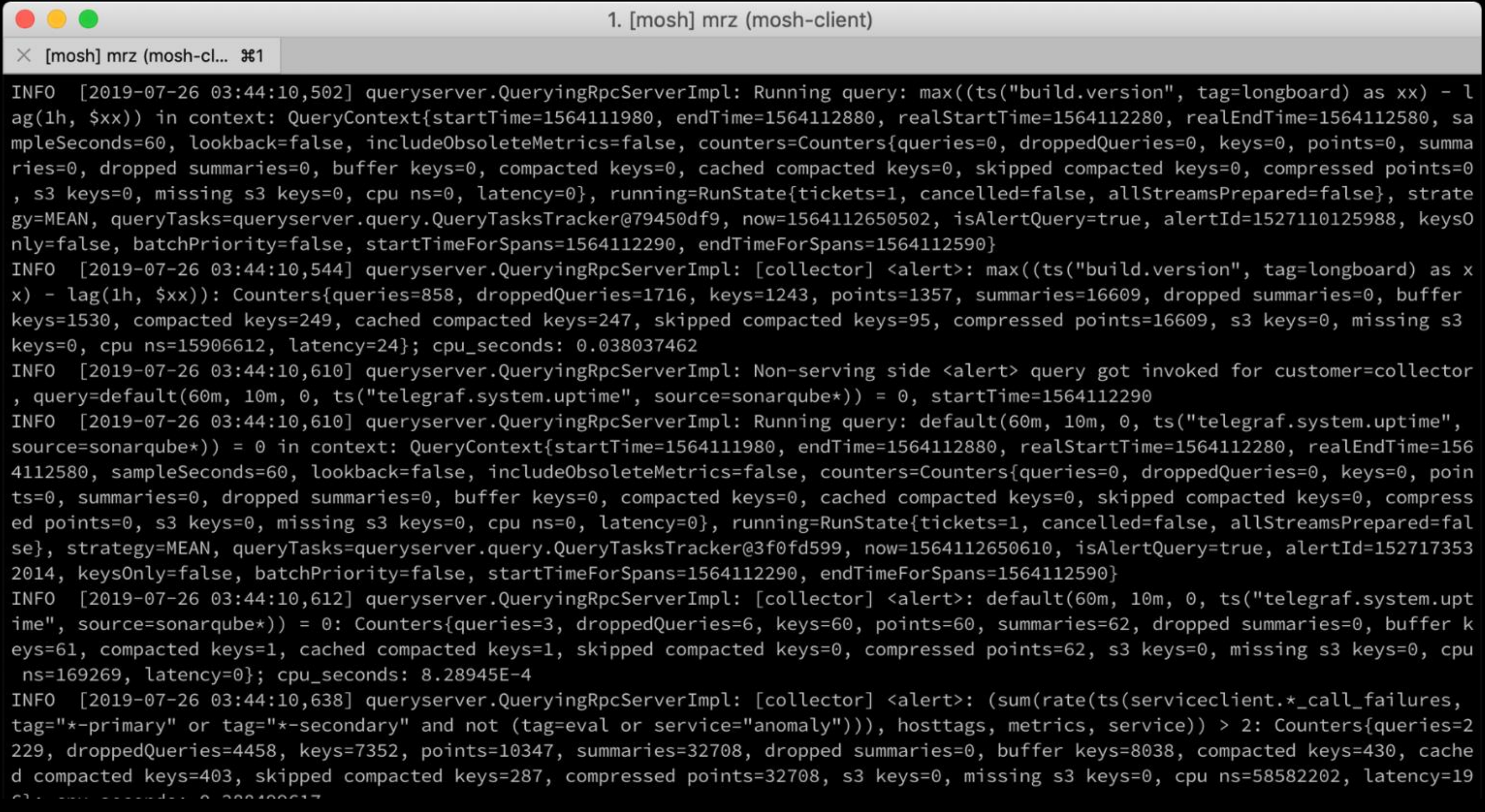
> Process: source = wavefront-tracing-example

∨ Logs (1)

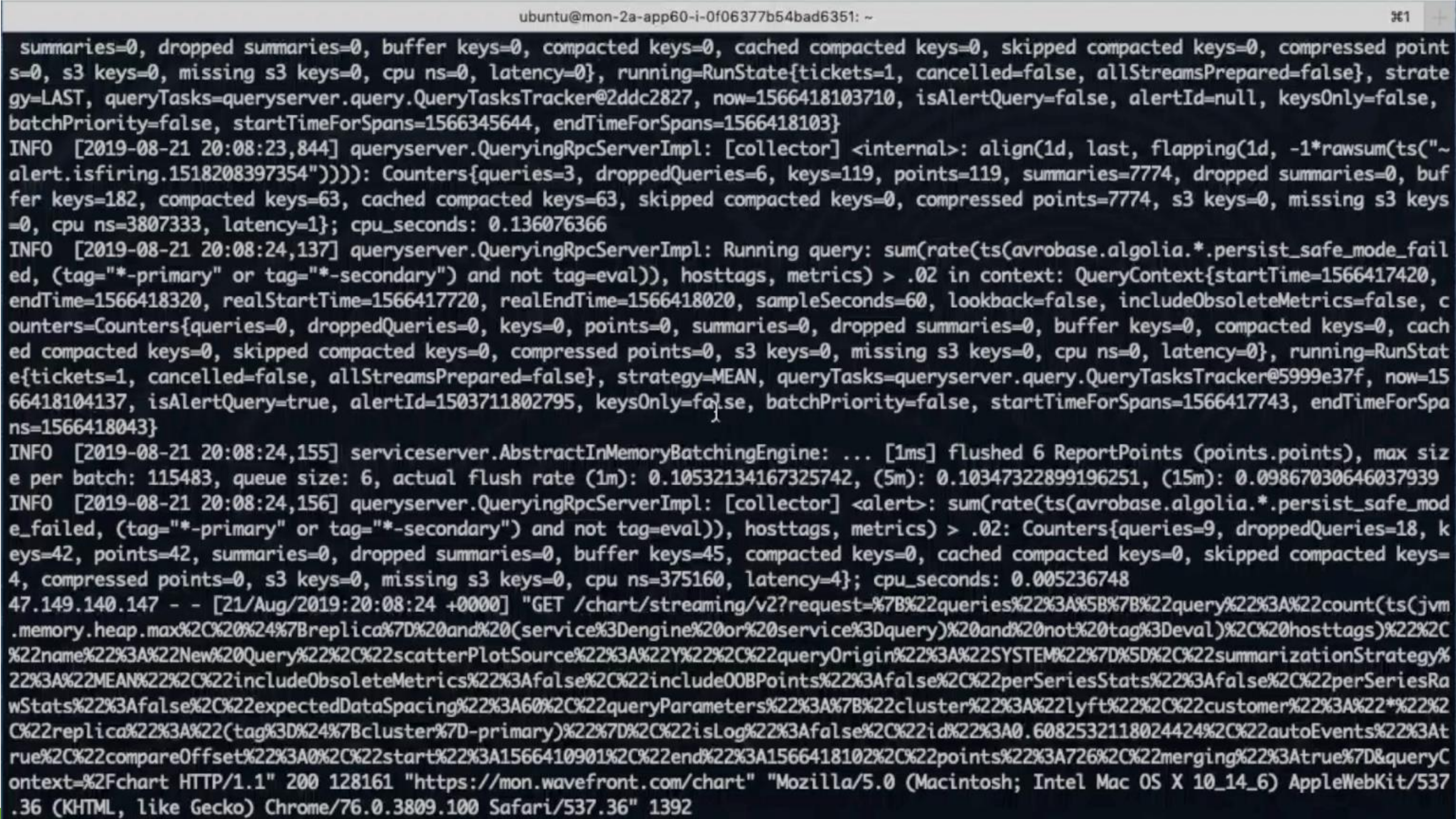
∨ July 18, 2019 4:33:12 PM - (13.05ms after trace start)

```
error.kind  "Exception"
event       "error"
message     "Failed to dispatch shirts!"
stack       "BeachShirts.Delivery: Oh no! ---> Syste
               at BeachShirts.Delivery.Controllers
               at BeachShirts.Delivery.Controlle
```

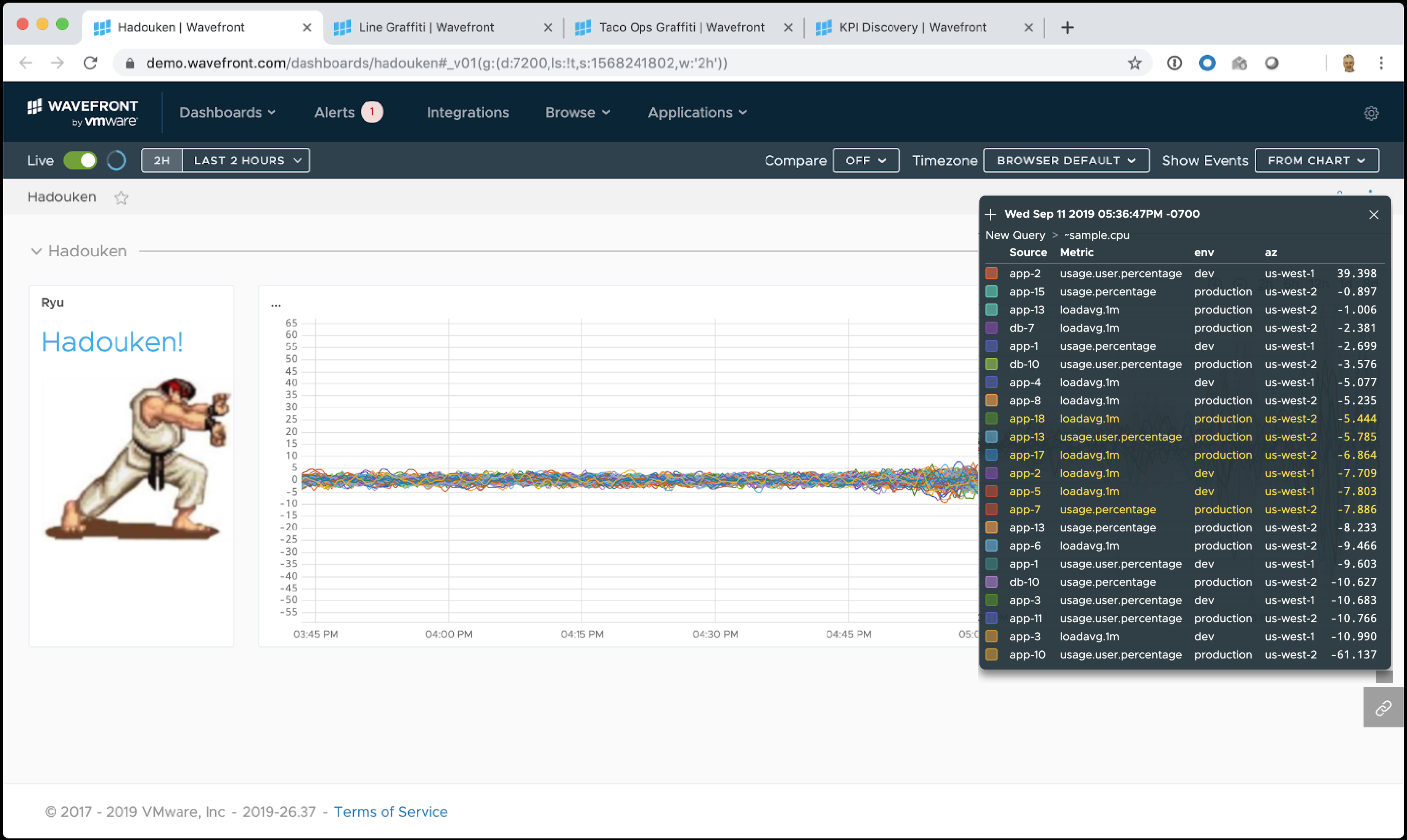
imestamps are relative to the start time of the full trac



```
INFO  [2019-07-26 03:44:10,502] queryserver.QueryingRpcServerImpl: Running query: max((ts("build.version", tag=longboard) as xx) - l
ag(1h, $xx)) in context: QueryContext{startTime=1564111980, endTime=1564112880, realStartTime=1564112280, realEndTime=1564112580, sa
mpleSeconds=60, lookback=false, includeObsoleteMetrics=false, counters=Counters{queries=0, droppedQueries=0, keys=0, points=0, summa
ries=0, dropped summaries=0, buffer keys=0, compacted keys=0, cached compacted keys=0, skipped compacted keys=0, compressed points=0
, s3 keys=0, missing s3 keys=0, cpu ns=0, latency=0}, running=RunState{tickets=1, cancelled=false, allStreamsPrepared=false}, strate
gy=MEAN, queryTasks=queryserver.query.QueryTasksTracker@79450df9, now=1564112650502, isAlertQuery=true, alertId=1527110125988, keysO
nly=false, batchPriority=false, startTimeForSpans=1564112290, endTimeForSpans=1564112590}
INFO  [2019-07-26 03:44:10,544] queryserver.QueryingRpcServerImpl: [collector] <alert>: max((ts("build.version", tag=longboard) as x
x) - lag(1h, $xx)): Counters{queries=858, droppedQueries=1716, keys=1243, points=1357, summaries=16609, dropped summaries=0, buffer
keys=1530, compacted keys=249, cached compacted keys=247, skipped compacted keys=95, compressed points=16609, s3 keys=0, missing s3
keys=0, cpu ns=15906612, latency=24}; cpu_seconds: 0.038037462
INFO  [2019-07-26 03:44:10,610] queryserver.QueryingRpcServerImpl: Non-serving side <alert> query got invoked for customer=collector
, query=default(60m, 10m, 0, ts("telegraf.system.uptime", source=sonarqube*)) = 0, startTime=1564112290
INFO  [2019-07-26 03:44:10,610] queryserver.QueryingRpcServerImpl: Running query: default(60m, 10m, 0, ts("telegraf.system.uptime",
source=sonarqube*)) = 0 in context: QueryContext{startTime=1564111980, endTime=1564112880, realStartTime=1564112280, realEndTime=156
4112580, sampleSeconds=60, lookback=false, includeObsoleteMetrics=false, counters=Counters{queries=0, droppedQueries=0, keys=0, poin
ts=0, summaries=0, dropped summaries=0, buffer keys=0, compacted keys=0, cached compacted keys=0, skipped compacted keys=0, compress
ed points=0, s3 keys=0, missing s3 keys=0, cpu ns=0, latency=0}, running=RunState{tickets=1, cancelled=false, allStreamsPrepared=fal
se}, strategy=MEAN, queryTasks=queryserver.query.QueryTasksTracker@3f0fd599, now=1564112650610, isAlertQuery=true, alertId=152717353
2014, keysOnly=false, batchPriority=false, startTimeForSpans=1564112290, endTimeForSpans=1564112590}
INFO  [2019-07-26 03:44:10,612] queryserver.QueryingRpcServerImpl: [collector] <alert>: default(60m, 10m, 0, ts("telegraf.system.upt
ime", source=sonarqube*)) = 0: Counters{queries=3, droppedQueries=6, keys=60, points=60, summaries=62, dropped summaries=0, buffer k
eys=61, compacted keys=1, cached compacted keys=1, skipped compacted keys=0, compressed points=62, s3 keys=0, missing s3 keys=0, cpu
 ns=169269, latency=0}; cpu_seconds: 8.28945E-4
INFO  [2019-07-26 03:44:10,638] queryserver.QueryingRpcServerImpl: [collector] <alert>: (sum(rate(ts(serviceclient.*_call_failures,
tag="*-primary" or tag="*-secondary" and not (tag=eval or service="anomaly"))), hosttags, metrics, service)) > 2: Counters{queries=2
229, droppedQueries=4458, keys=7352, points=10347, summaries=32708, dropped summaries=0, buffer keys=8038, compacted keys=430, cache
d compacted keys=403, skipped compacted keys=287, compressed points=32708, s3 keys=0, missing s3 keys=0, cpu ns=58582202, latency=19
6}; cpu_seconds: 0.288409617
```



```
 summaries=0, dropped summaries=0, buffer keys=0, compacted keys=0, cached compacted keys=0, skipped compacted keys=0, compressed point
s=0, s3 keys=0, missing s3 keys=0, cpu ns=0, latency=0}, running=RunState{tickets=1, cancelled=false, allStreamsPrepared=false}, strate
gy=LAST, queryTasks=queryserver.query.QueryTasksTracker@2ddc2827, now=1566418103710, isAlertQuery=false, alertId=null, keysOnly=false,
batchPriority=false, startTimeForSpans=1566345644, endTimeForSpans=1566418103}
INFO  [2019-08-21 20:08:23,844] queryserver.QueryingRpcServerImpl: [collector] <internal>: align(1d, last, flapping(1d, -1*rawsum(ts("~
alert.isfiring.1518208397354")))): Counters{queries=3, droppedQueries=6, keys=119, points=119, summaries=7774, dropped summaries=0, buf
fer keys=182, compacted keys=63, cached compacted keys=63, skipped compacted keys=0, compressed points=7774, s3 keys=0, missing s3 keys
=0, cpu ns=3807333, latency=1}; cpu_seconds: 0.136076366
INFO  [2019-08-21 20:08:24,137] queryserver.QueryingRpcServerImpl: Running query: sum(rate(ts(avrobase.algolia.*.persist_safe_mode_fail
ed, (tag="*-primary" or tag="*-secondary") and not tag=eval)), hosttags, metrics) > .02 in context: QueryContext{startTime=1566417420,
endTime=1566418320, realStartTime=1566417720, realEndTime=1566418020, sampleSeconds=60, lookback=false, includeObsoleteMetrics=false, c
ounters=Counters{queries=0, droppedQueries=0, keys=0, points=0, summaries=0, dropped summaries=0, buffer keys=0, compacted keys=0, cach
ed compacted keys=0, skipped compacted keys=0, compressed points=0, s3 keys=0, missing s3 keys=0, cpu ns=0, latency=0}, running=RunStat
e{tickets=1, cancelled=false, allStreamsPrepared=false}, strategy=MEAN, queryTasks=queryserver.query.QueryTasksTracker@5999e37f, now=15
66418104137, isAlertQuery=true, alertId=1503711802795, keysOnly=false, batchPriority=false, startTimeForSpans=1566417743, endTimeForSpa
ns=1566418043}
INFO  [2019-08-21 20:08:24,155] serviceserver.AbstractInMemoryBatchingEngine: ... [1ms] flushed 6 ReportPoints (points.points), max siz
e per batch: 115483, queue size: 6, actual flush rate (1m): 0.10532134167325742, (5m): 0.10347322899196251, (15m): 0.09867030646037939
INFO  [2019-08-21 20:08:24,156] queryserver.QueryingRpcServerImpl: [collector] <alert>: sum(rate(ts(avrobase.algolia.*.persist_safe_mod
e_failed, (tag="*-primary" or tag="*-secondary") and not tag=eval)), hosttags, metrics) > .02: Counters{queries=9, droppedQueries=18, k
eys=42, points=42, summaries=0, dropped summaries=0, buffer keys=45, compacted keys=0, cached compacted keys=0, skipped compacted keys=
4, compressed points=0, s3 keys=0, missing s3 keys=0, cpu ns=375160, latency=4}; cpu_seconds: 0.005236748
47.149.140.147 - - [21/Aug/2019:20:08:24 +0000] "GET /chart/streaming/v2?request=%7B%22queries%22%3A%5B%7B%22query%22%3A%22count(ts(jvm
.memory.heap.max%2C%20%24%7Breplica%7D%20and%20(service%3Dengine%20or%20service%3Dquery)%20and%20not%20tag%3Deval)%2C%20hosttags)%22%2C
%22name%22%3A%22New%20Query%22%2C%22scatterPlotSource%22%3A%22Y%22%2C%22queryOrigin%22%3A%22SYSTEM%22%7D%5D%2C%22summarizationStrategy%
22%3A%22MEAN%22%2C%22includeObsoleteMetrics%22%3Afalse%2C%22includeOOBPoints%22%3Afalse%2C%22perSeriesStats%22%3Afalse%2C%22perSeriesRa
wStats%22%3Afalse%2C%22expectedDataSpacing%22%3A60%2C%22queryParameters%22%3A%7B%22cluster%22%3A%22lyft%22%2C%22customer%22%3A%22*%22%2
C%22replica%22%3A%22(tag%3D%24%7Bcluster%7D-primary)%22%7D%2C%22isLog%22%3Afalse%2C%22id%22%3A0.6082532118024424%2C%22autoEvents%22%3At
rue%2C%22compareOffset%22%3A0%2C%22start%22%3A1566410901%2C%22end%22%3A1566418102%2C%22points%22%3A726%2C%22merging%22%3Atrue%7D&queryC
ontext=%2Fchart HTTP/1.1" 200 128161 "https://mon.wavefront.com/chart" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537
.36 (KHTML, like Gecko) Chrome/76.0.3809.100 Safari/537.36" 1392
```



demo.wavefront.com/dashboards/hadouken#_v01(g:(d:7200,ls:!t,s:1568241802,w:'2h'))

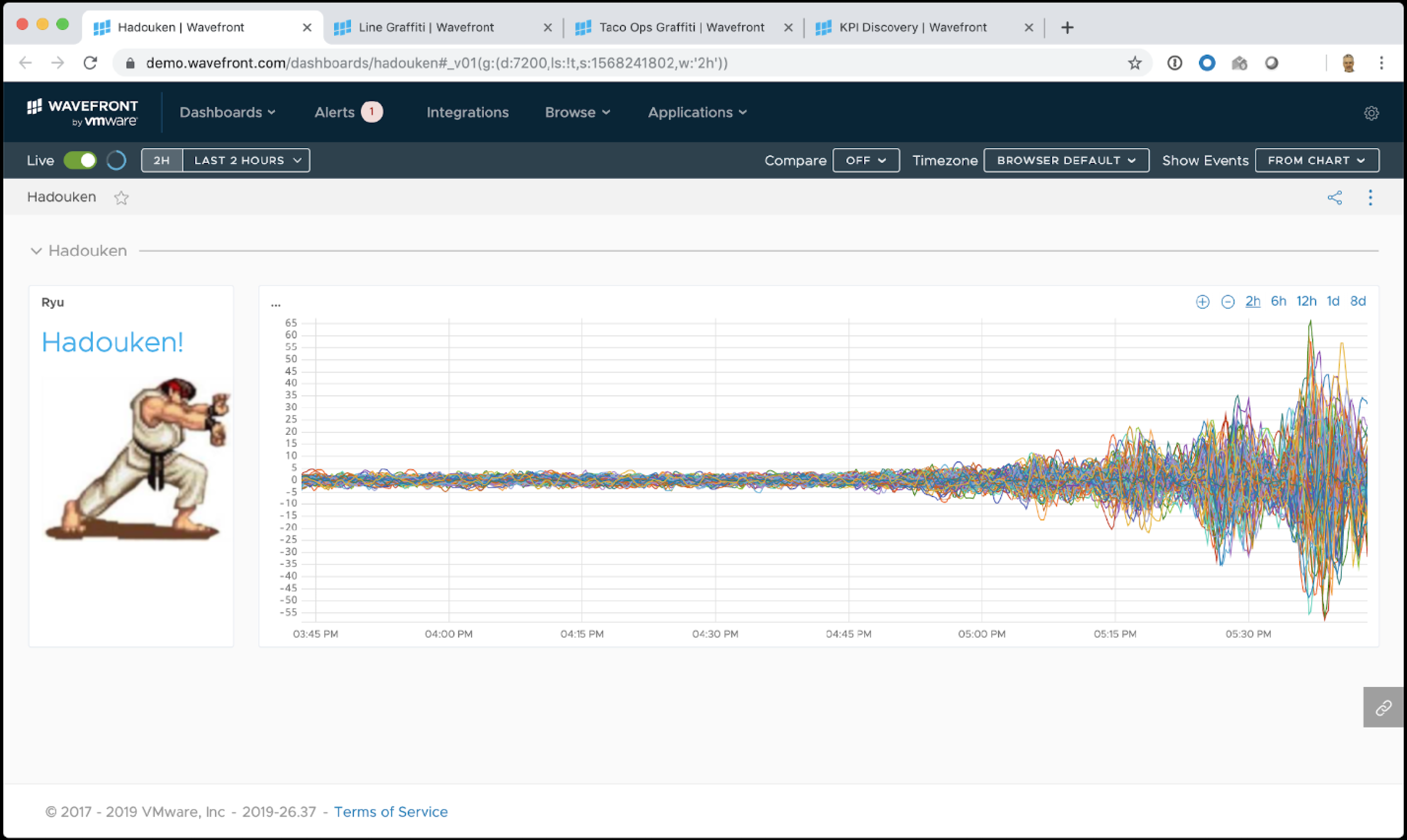
Hadouken!

Wed Sep 11 2019 05:36:47PM -0700

New Query > ~sample.cpu

	Source	Metric	env	az	
■	app-2	usage.user.percentage	dev	us-west-1	39.398
■	app-15	usage.percentage	production	us-west-2	-0.897
■	app-13	loadavg.1m	production	us-west-2	-1.006
■	db-7	loadavg.1m	production	us-west-2	-2.381
■	app-1	usage.percentage	dev	us-west-1	-2.699
■	db-10	usage.user.percentage	production	us-west-2	-3.576
■	app-4	loadavg.1m	dev	us-west-1	-5.077
■	app-8	loadavg.1m	production	us-west-2	-5.235
■	app-18	loadavg.1m	production	us-west-2	-5.444
■	app-13	usage.user.percentage	production	us-west-2	-5.785
■	app-17	loadavg.1m	production	us-west-2	-6.864
■	app-2	loadavg.1m	dev	us-west-1	-7.709
■	app-5	loadavg.1m	dev	us-west-1	-7.803
■	app-7	usage.percentage	production	us-west-2	-7.886
■	app-13	usage.percentage	production	us-west-2	-8.233
■	app-6	loadavg.1m	production	us-west-2	-9.466
■	app-1	usage.user.percentage	dev	us-west-1	-9.603
■	db-10	usage.percentage	production	us-west-2	-10.627
■	app-3	usage.user.percentage	dev	us-west-1	-10.683
■	app-11	usage.user.percentage	production	us-west-2	-10.766
■	app-3	loadavg.1m	dev	us-west-1	-10.990
■	app-10	usage.user.percentage	production	us-west-2	-61.137

© 2017 - 2019 VMware, Inc - 2019-26.37 - Terms of Service
]

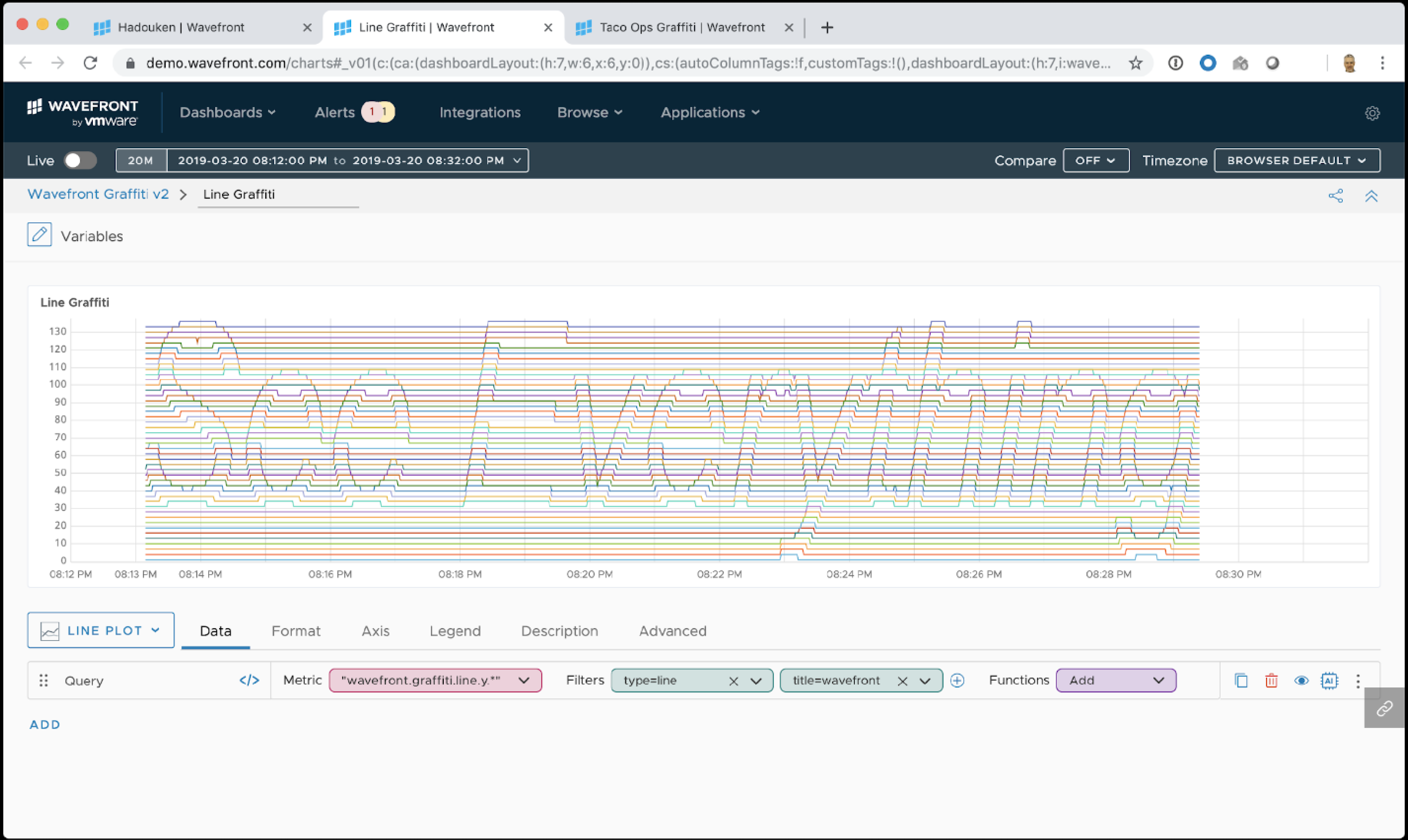


demo.wavefront.com/dashboards/hadouken#_v01(g:(d:7200,ls:!t,s:1568241802,w:'2h'))

Live 2H LAST 2 HOURS

Compare OFF Timezone BROWSER DEFAULT Show Events FROM CHART

© 2017 - 2019 VMware, Inc - 2019-26.37 - Terms of Service

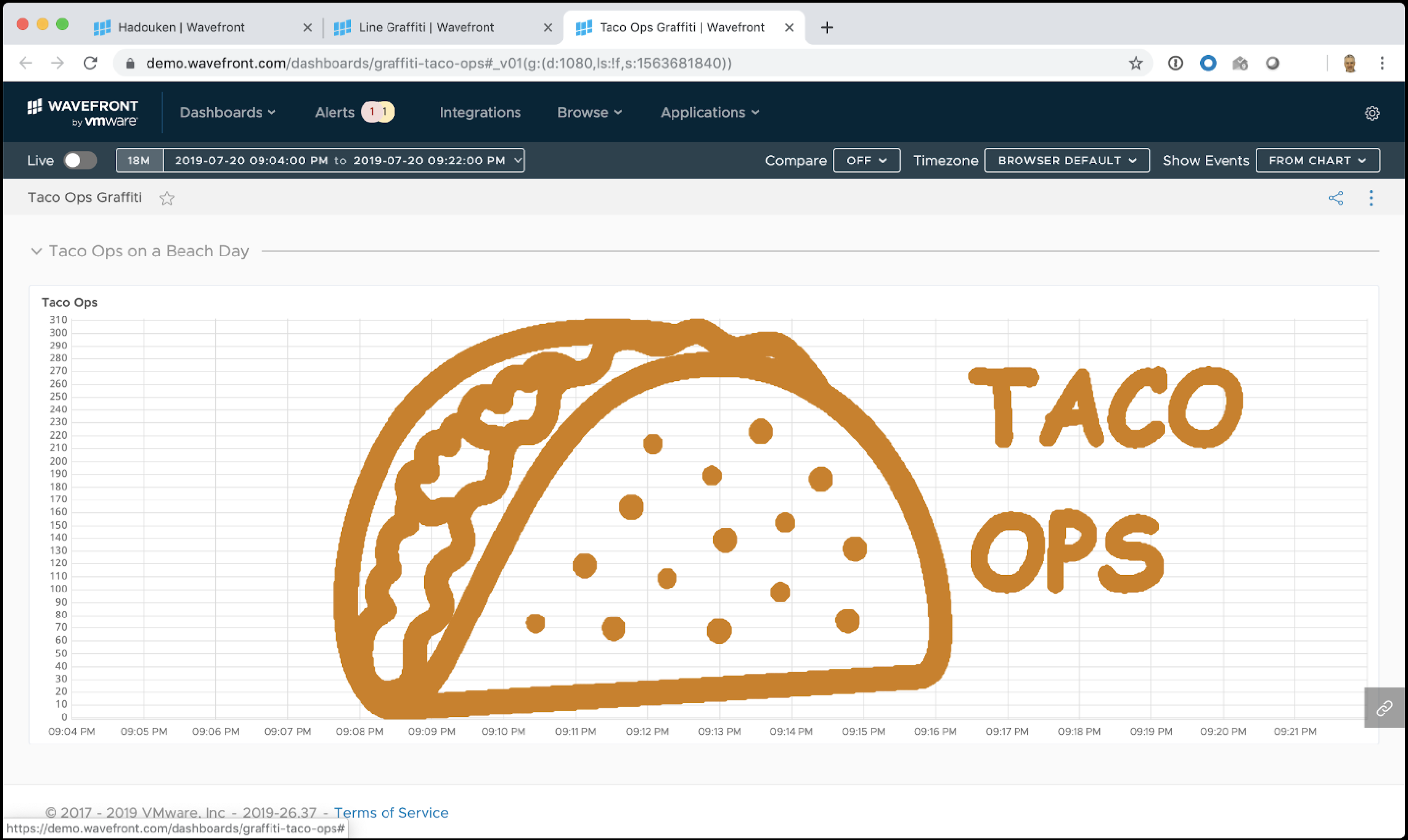


demo.wavefront.com/charts#_v01(c:(ca:(dashboardLayout:(h:7,w:6,x:6,y:0)),cs:(autoColumnTags:!f,customTags:!(),dashboardLayout:(h:7,i:wave...

20M 2019-03-20 08:12:00 PM to 2019-03-20 08:32:00 PM

Metric "wavefront.graffiti.line.y.*"

Filters type=line title=wavefront



demo.wavefront.com/dashboards/graffiti-taco-ops#_v01(g:(d:1080,ls:!f,s:1563681840))

Live ○ 18M 2019-07-20 09:04:00 PM to 2019-07-20 09:22:00 PM

Compare OFF Timezone BROWSER DEFAULT Show Events FROM CHART

Taco Ops Graffiti ☆

Taco Ops on a Beach Day

Taco Ops

© 2017 - 2019 VMware, Inc - 2019-26.37 - Terms of Service

https://demo.wavefront.com/dashboards/graffiti-taco-ops#

Observability for VMware Cloud Services

100+
Application Services

500+
Users

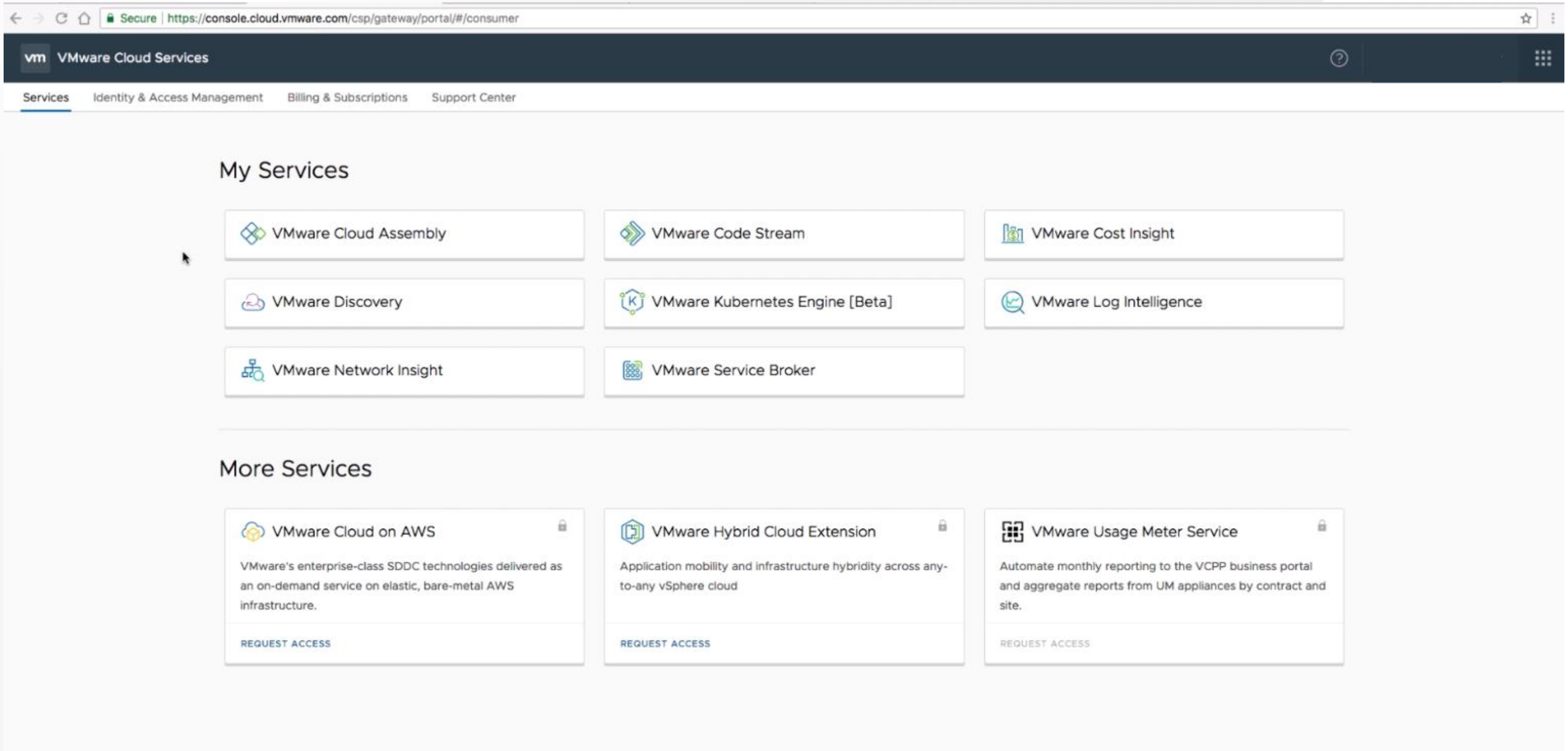
5,000+
Containers

600+
Dashboards

12+
Kubernetes Clusters
US, UK, Tokyo Regions

700+
Alerts

All the Things. On Kubernetes. And Ocean.



Secure | https://console.cloud.vmware.com/csp/gateway/portal/#/consumer

VMware Cloud Services

Services | Identity & Access Management | Billing & Subscriptions | Support Center

My Services

- VMware Cloud Assembly
- VMware Code Stream
- VMware Cost Insight
- VMware Discovery
- VMware Kubernetes Engine [Beta]
- VMware Log Intelligence
- VMware Network Insight
- VMware Service Broker

More Services

VMware Cloud on AWS

VMware's enterprise-class SDDC technologies delivered as an on-demand service on elastic, bare-metal AWS infrastructure.

REQUEST ACCESS

VMware Hybrid Cloud Extension

Application mobility and infrastructure hybridity across any-to-any vSphere cloud

REQUEST ACCESS

VMware Usage Meter Service

Automate monthly reporting to the VCPP business portal and aggregate reports from UM appliances by contract and site.

REQUEST ACCESS

WAVEFRONT

♡

spotinst

Wavefront & Spotinst

Enhanced UX Dashboarding

Enterprise Usage Reporting

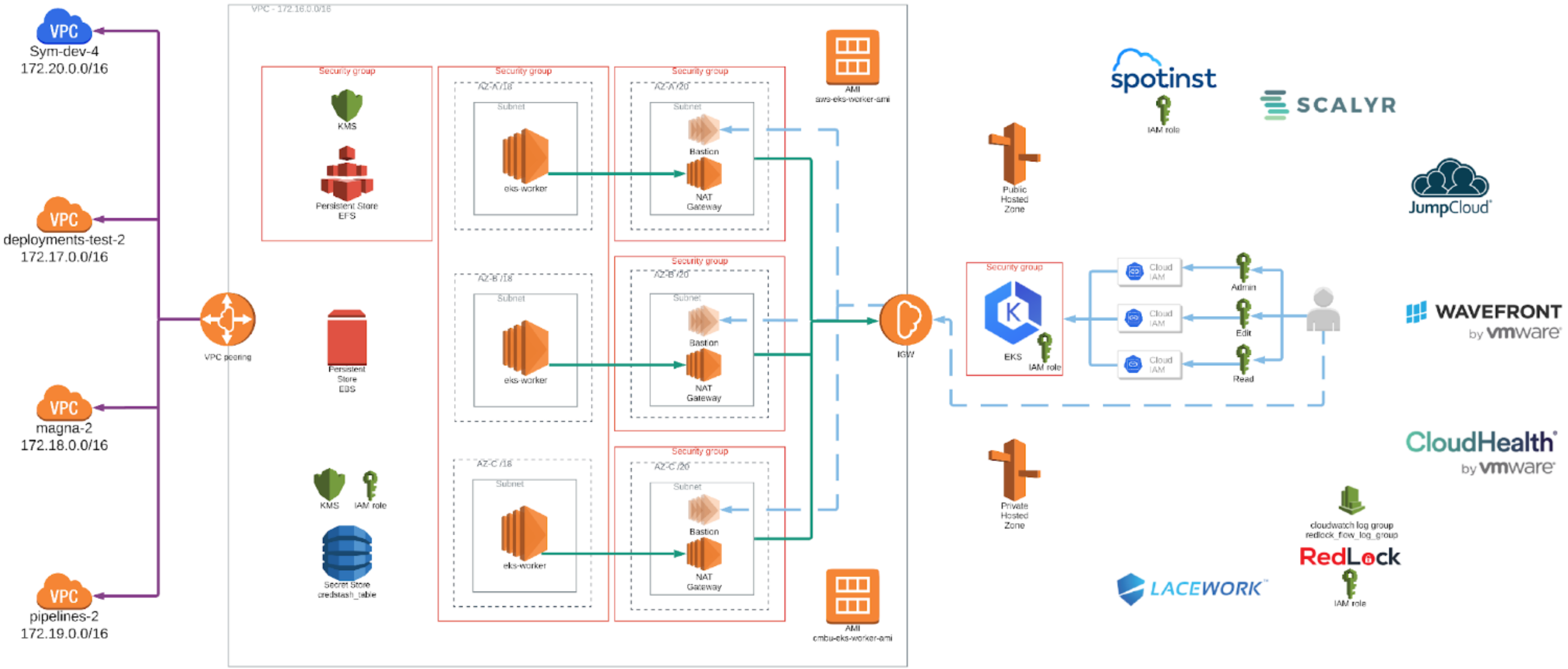
Automatic Kubernetes Observability

Strengthened Application Observability

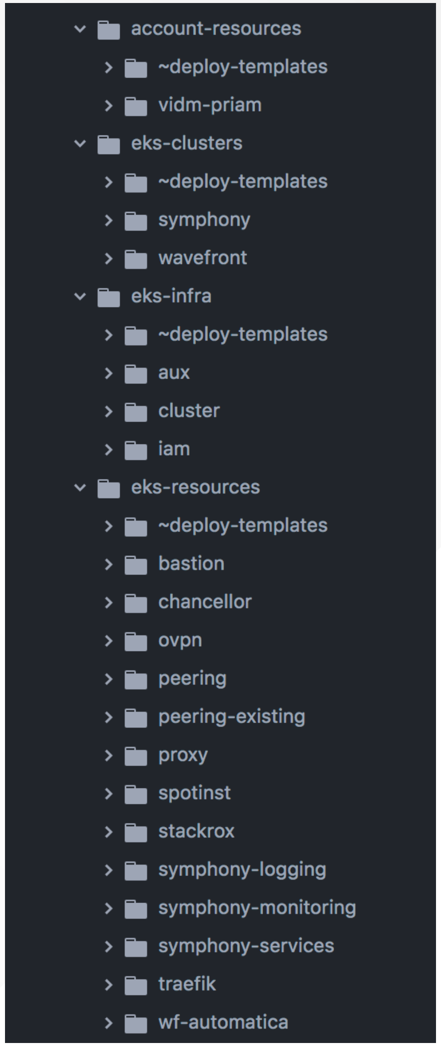
Demo

Architecture

sym-dev4-mgmt-2

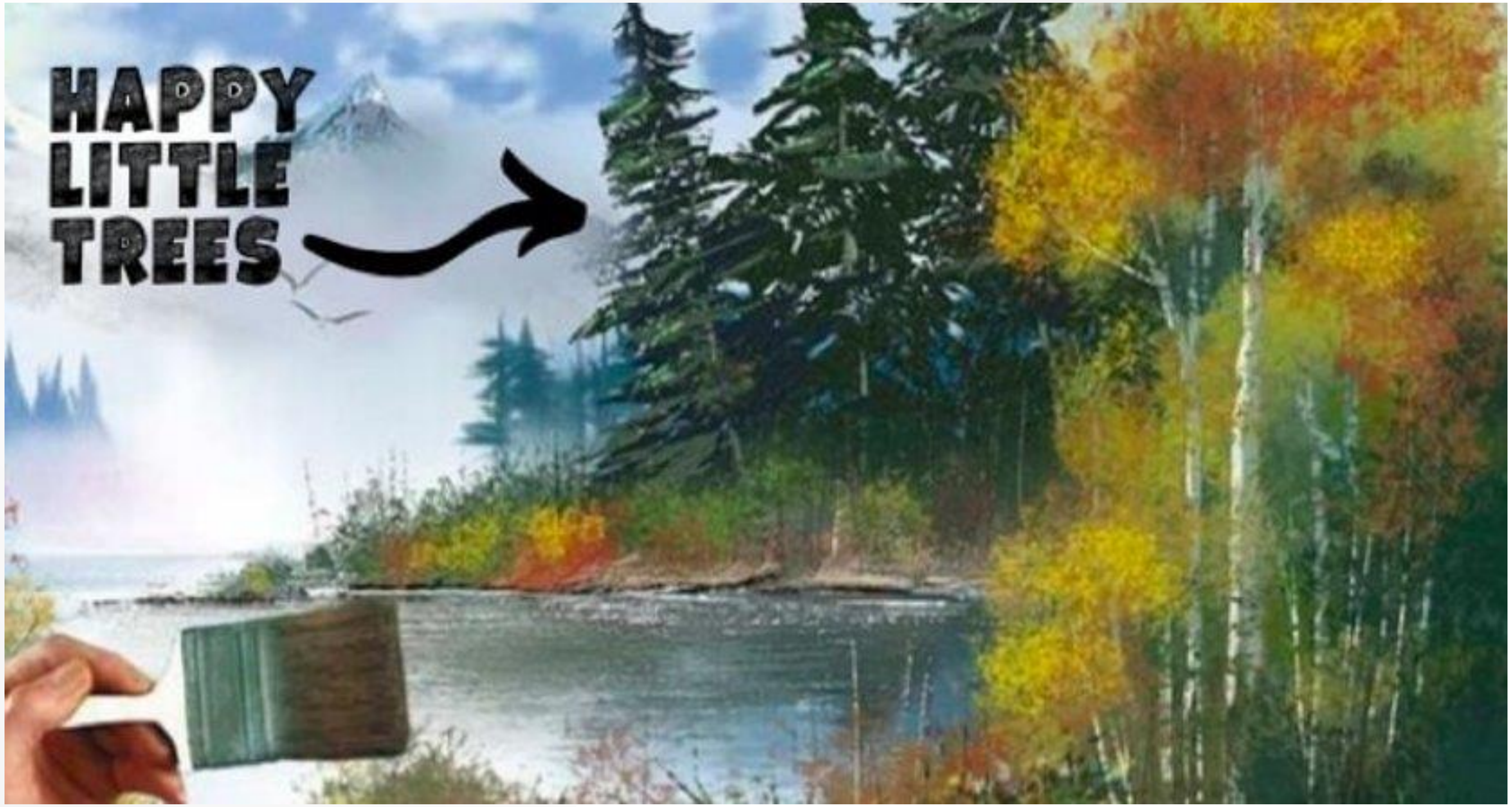


Code Structure

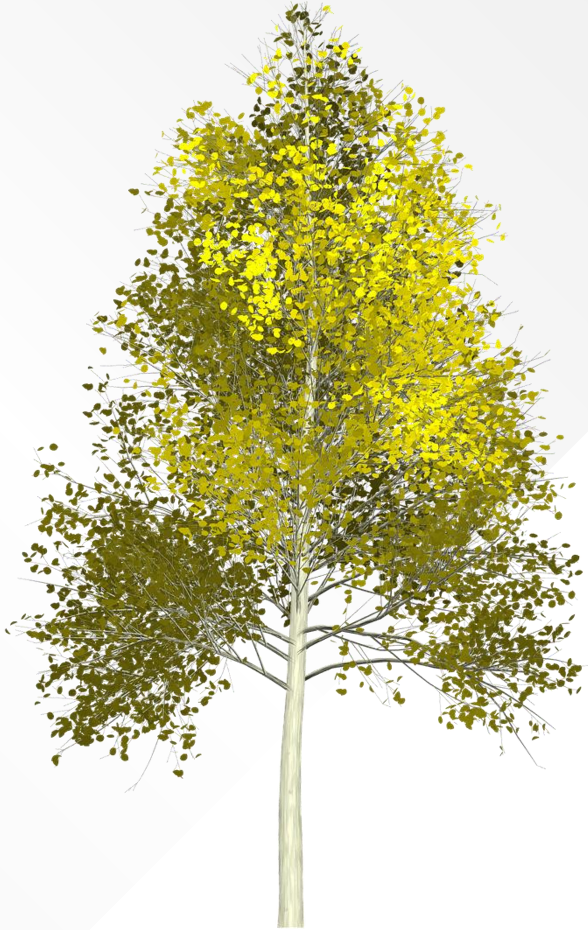


```
account-resources
  ~deploy-templates
  vidm-priam
eks-clusters
  ~deploy-templates
  symphony
  wavefront
eks-infra
  ~deploy-templates
  aux
  cluster
  iam
eks-resources
  ~deploy-templates
  bastion
  chancellor
  ovpn
  peering
  peering-existing
  proxy
  spotinst
  stackrox
  symphony-logging
  symphony-monitoring
  symphony-services
  traefik
  wf-automatica
```

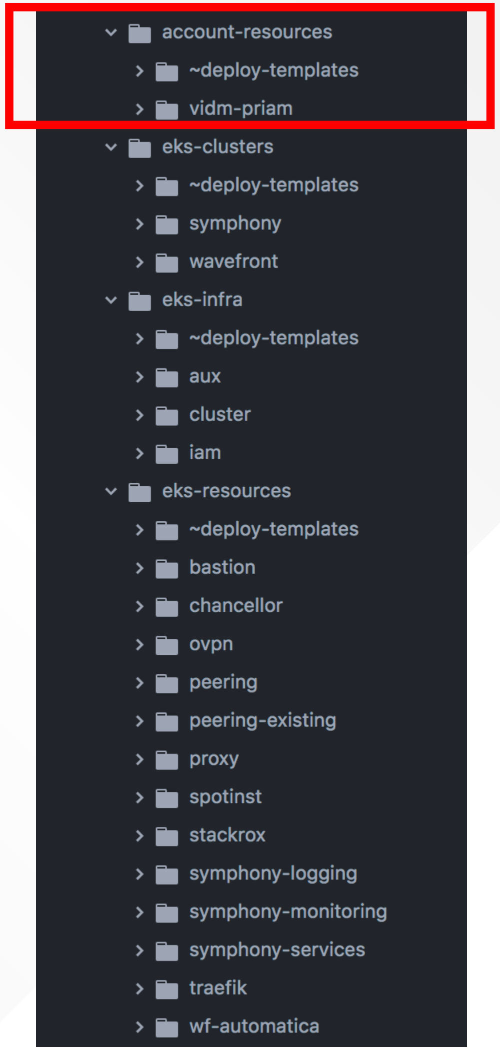
Code Structure



Code Structure



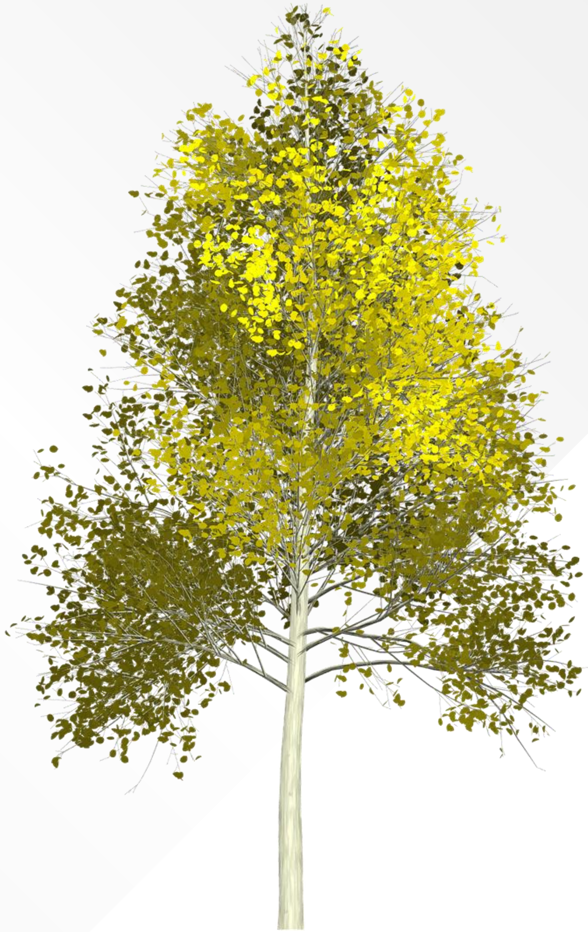
Code Structure

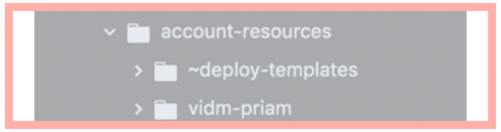


- account-resources
  - ~deploy-templates
  - vidm-priam
- eks-clusters
  - ~deploy-templates
  - symphony
  - wavefront
- eks-infra
  - ~deploy-templates
  - aux
  - cluster
  - iam
- eks-resources
  - ~deploy-templates
  - bastion
  - chancellor
  - ovpn
  - peering
  - peering-existing
  - proxy
  - spotinst
  - stackrox
  - symphony-logging
  - symphony-monitoring
  - symphony-services
  - traefik
  - wf-automatica

Run once per account
(required)

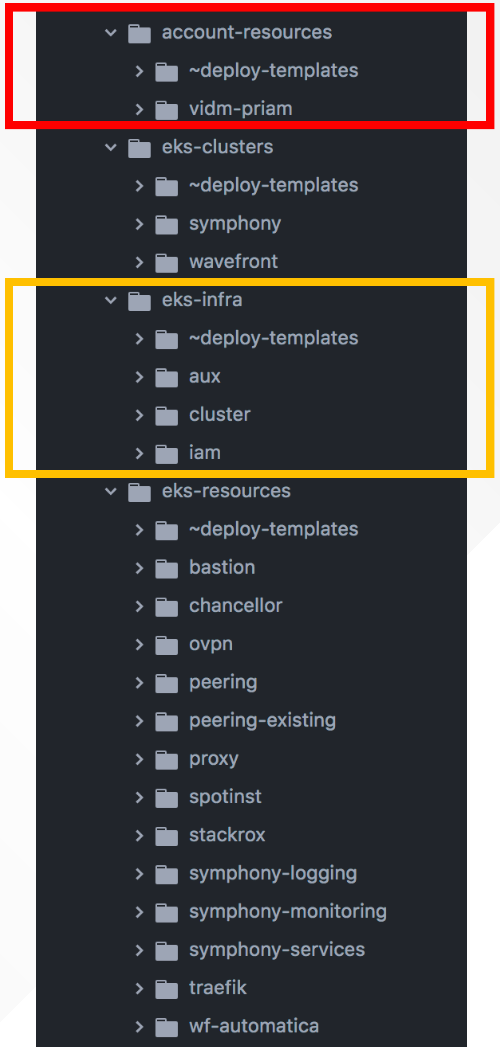
Code Structure





```
⌄ account-resources
  › ~deploy-templates
  › vidm-priam
```

Code Structure

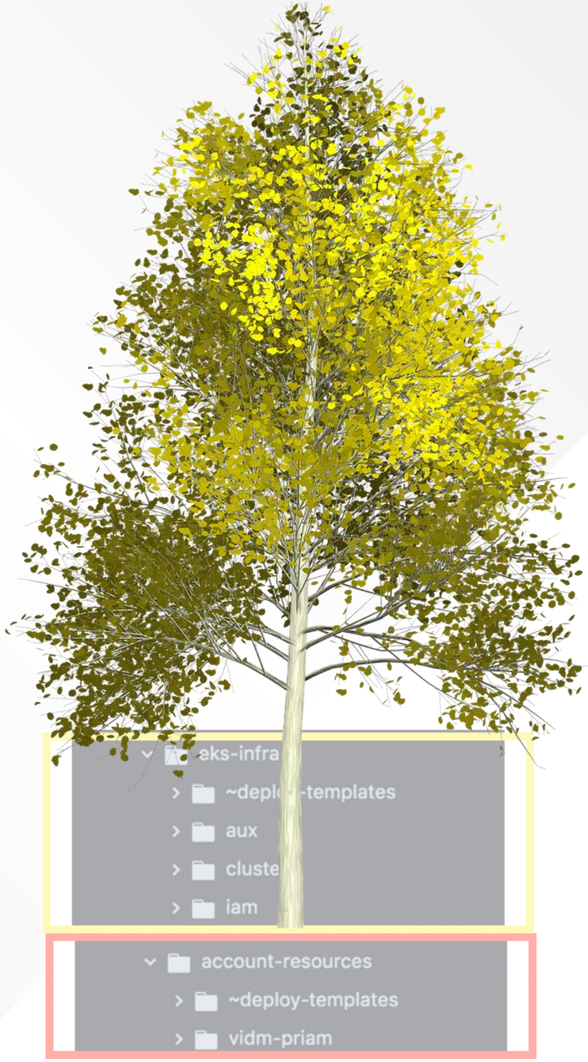


```
⌄ account-resources
  › ~deploy-templates
  › vidm-priam
⌄ eks-clusters
  › ~deploy-templates
  › symphony
  › wavefront
⌄ eks-infra
  › ~deploy-templates
  › aux
  › cluster
  › iam
⌄ eks-resources
  › ~deploy-templates
  › bastion
  › chancellor
  › ovpn
  › peering
  › peering-existing
  › proxy
  › spotinst
  › stackrox
  › symphony-logging
  › symphony-monitoring
  › symphony-services
  › traefik
  › wf-automatica
```

Run once per account
(required)

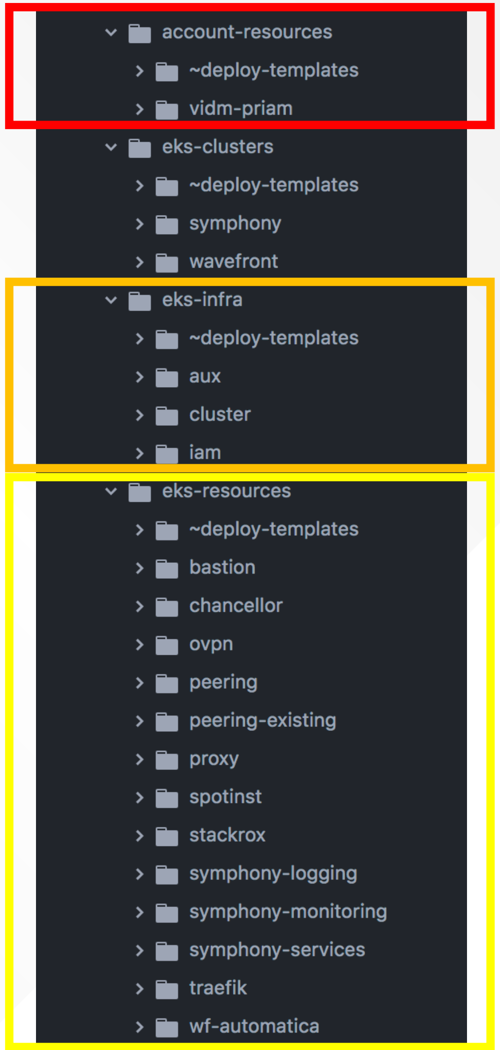
Run once per cluster
(required)

Code Structure



```
eks-infra
    ~deploy-templates
    aux
    cluster
    iam
account-resources
    ~deploy-templates
    vidm-priam
```

Code Structure



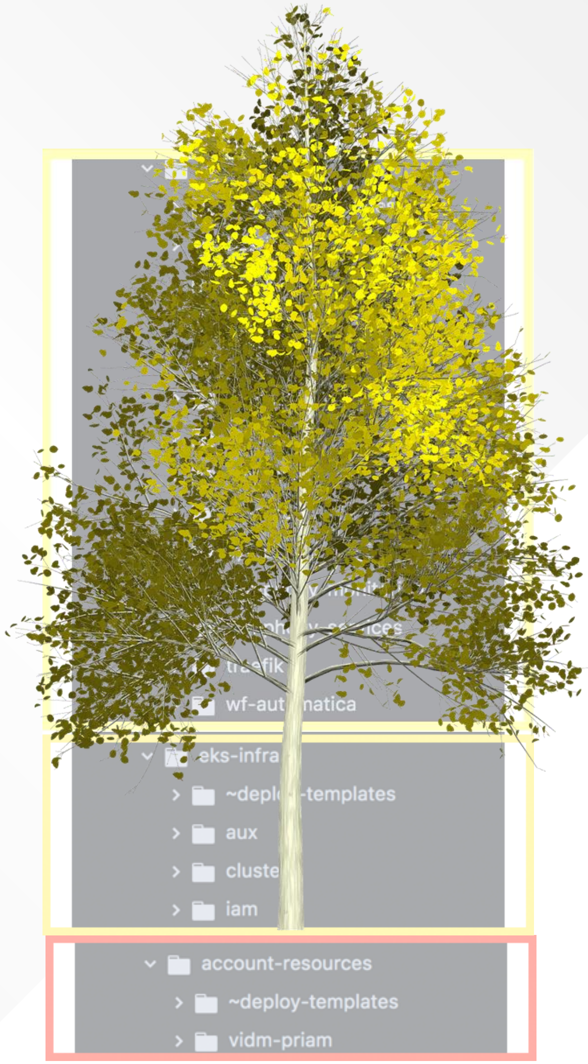
```
account-resources
  ~deploy-templates
  vidm-priam
eks-clusters
  ~deploy-templates
  symphony
  wavefront
eks-infra
  ~deploy-templates
  aux
  cluster
  iam
eks-resources
  ~deploy-templates
  bastion
  chancellor
  ovpn
  peering
  peering-existing
  proxy
  spotinst
  stackrox
  symphony-logging
  symphony-monitoring
  symphony-services
  traefik
  wf-automatica
```

Run once per account
(required)

Run once per cluster
(required)

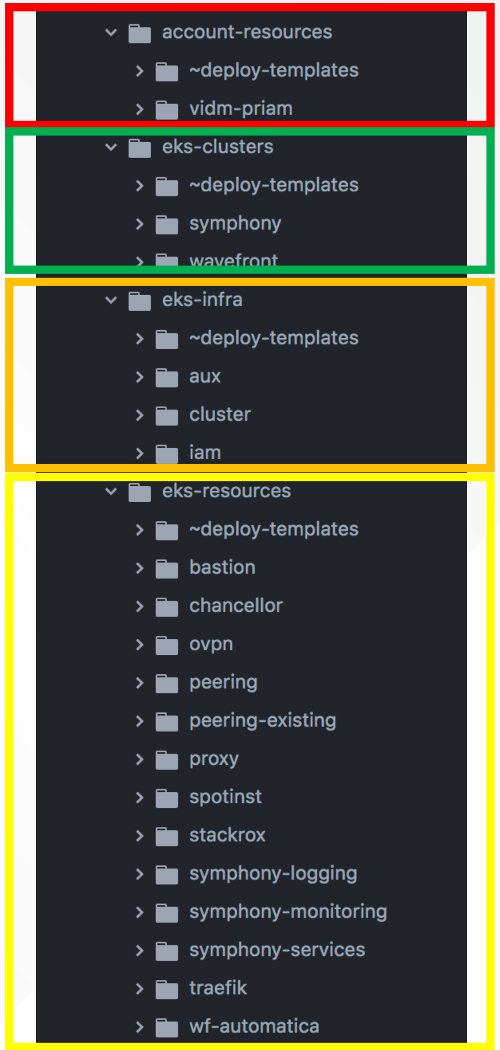
Run once per cluster
(optional)

Code Structure



```
    [illegible]
    [illegible]-services
    traefik
    wf-automatica
⌄ eks-infra
  › ~deploy-templates
  › aux
  › cluster
  › iam
⌄ account-resources
  › ~deploy-templates
  › vidm-priam
```

Code Structure



```
account-resources
    ~deploy-templates
    vidm-priam
eks-clusters
    ~deploy-templates
    symphony
    wavefront
eks-infra
    ~deploy-templates
    aux
    cluster
    iam
eks-resources
    ~deploy-templates
    bastion
    chancellor
    ovpn
    peering
    peering-existing
    proxy
    spotinst
    stackrox
    symphony-logging
    symphony-monitoring
    symphony-services
    traefik
    wf-automatica
```

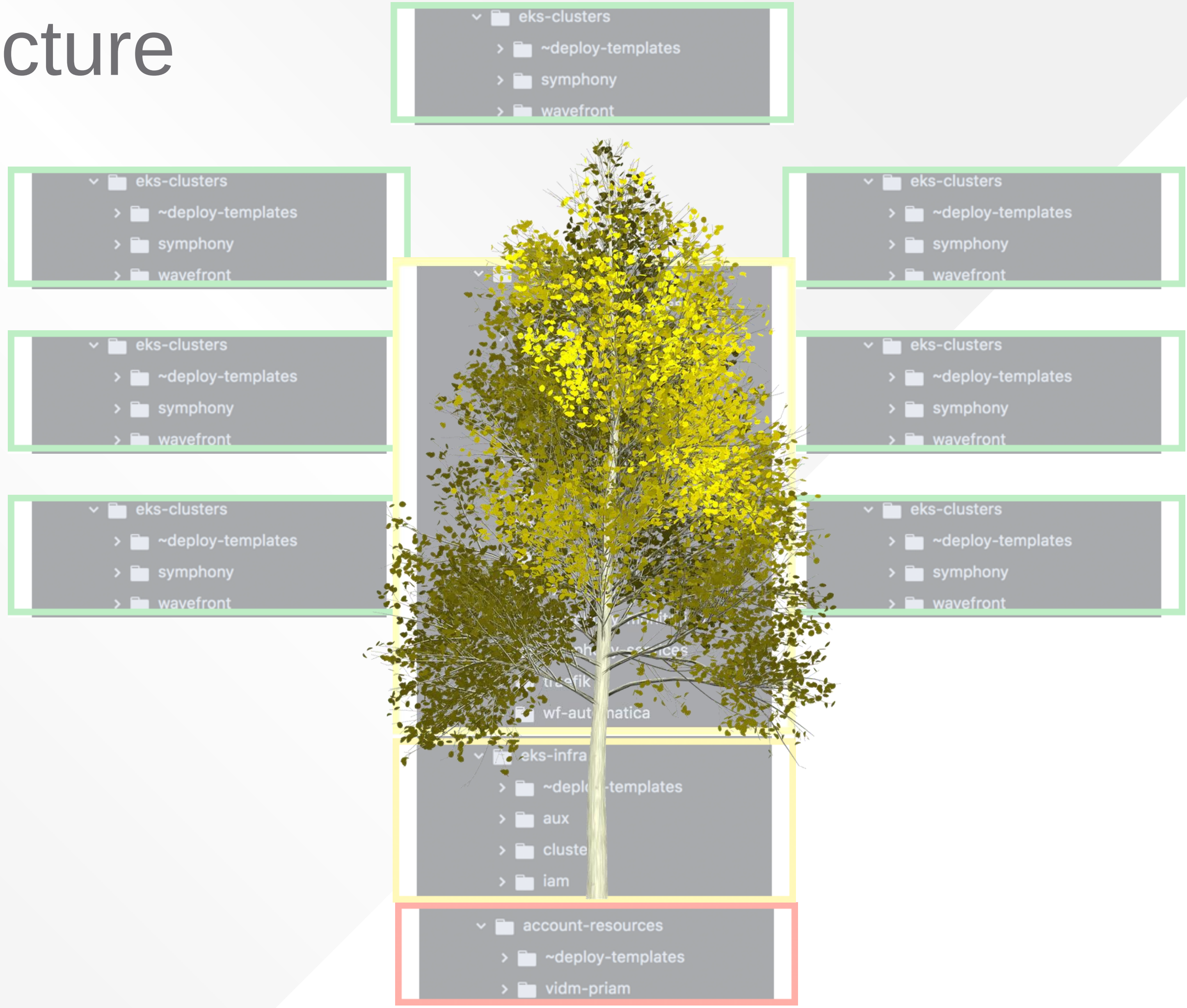
Run once per account
(required)

Cluster definitions
(required)

Cluster scaffolding
(required)

Cluster extensions
(optional)

Code Structure



```
eks-clusters
  ~deploy-templates
  symphony
  wavefront
```

```
eks-clusters
  ~deploy-templates
  symphony
  wavefront
```

```
eks-clusters
  ~deploy-templates
  symphony
  wavefront
```

```
eks-clusters
  ~deploy-templates
  symphony
  wavefront
```

```
eks-clusters
  ~deploy-templates
  symphony
  wavefront
```

```
eks-clusters
  ~deploy-templates
  symphony
  wavefront
```

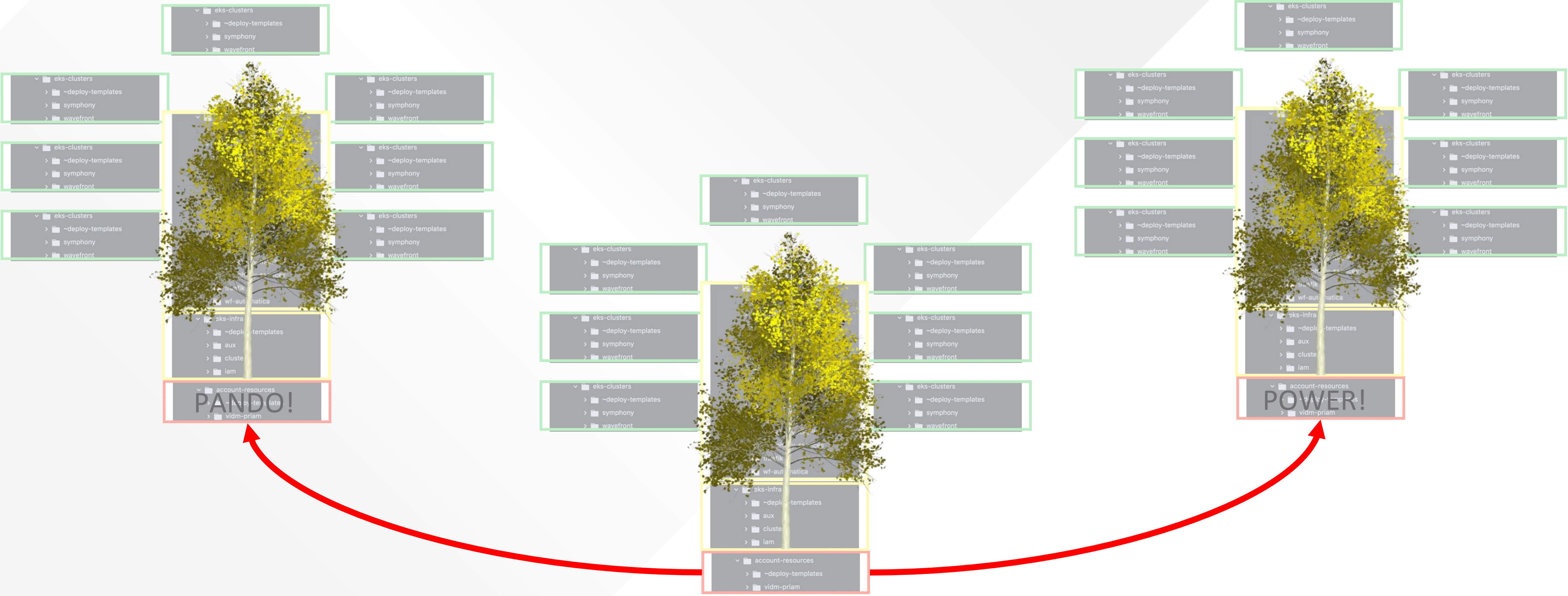
```
eks-clusters
  ~deploy-templates
  symphony
  wavefront
```

```
  symphony-services
  traefik
  wf-automatica
```

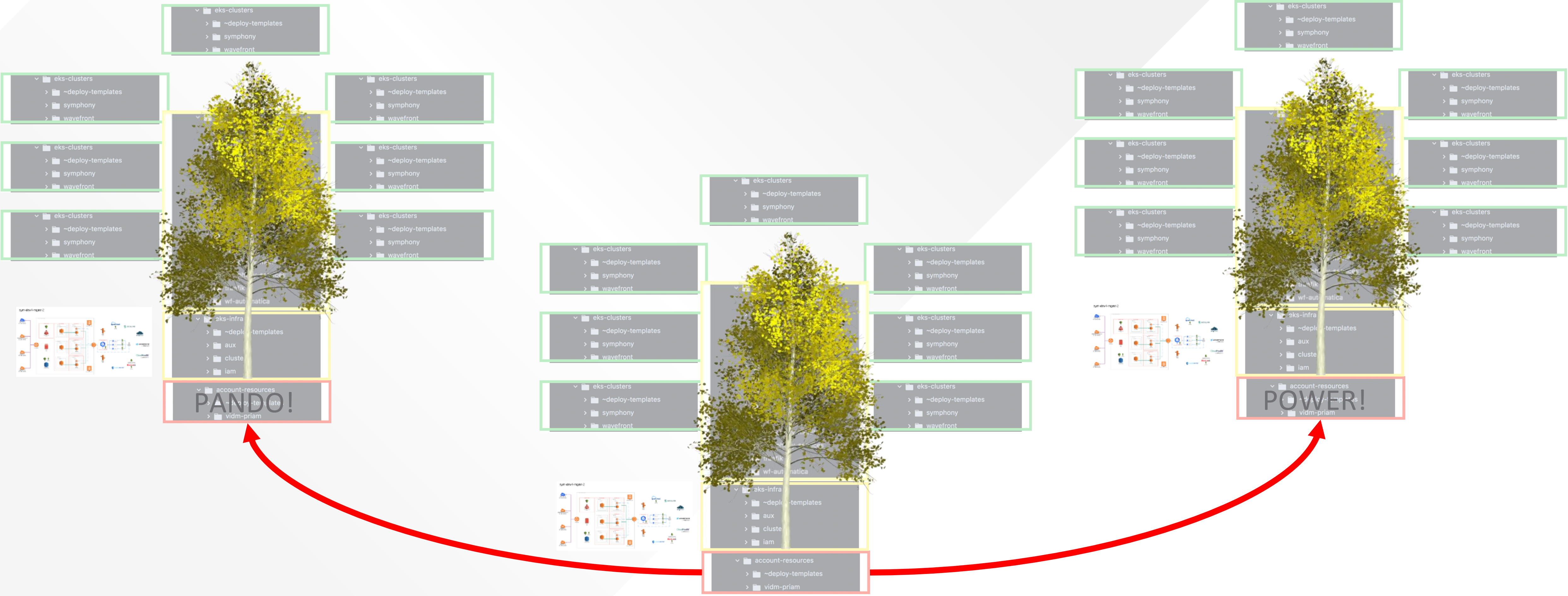
```
eks-infra
  ~deploy-templates
  aux
  cluste
  iam
```

```
account-resources
  ~deploy-templates
  vidm-priam
```

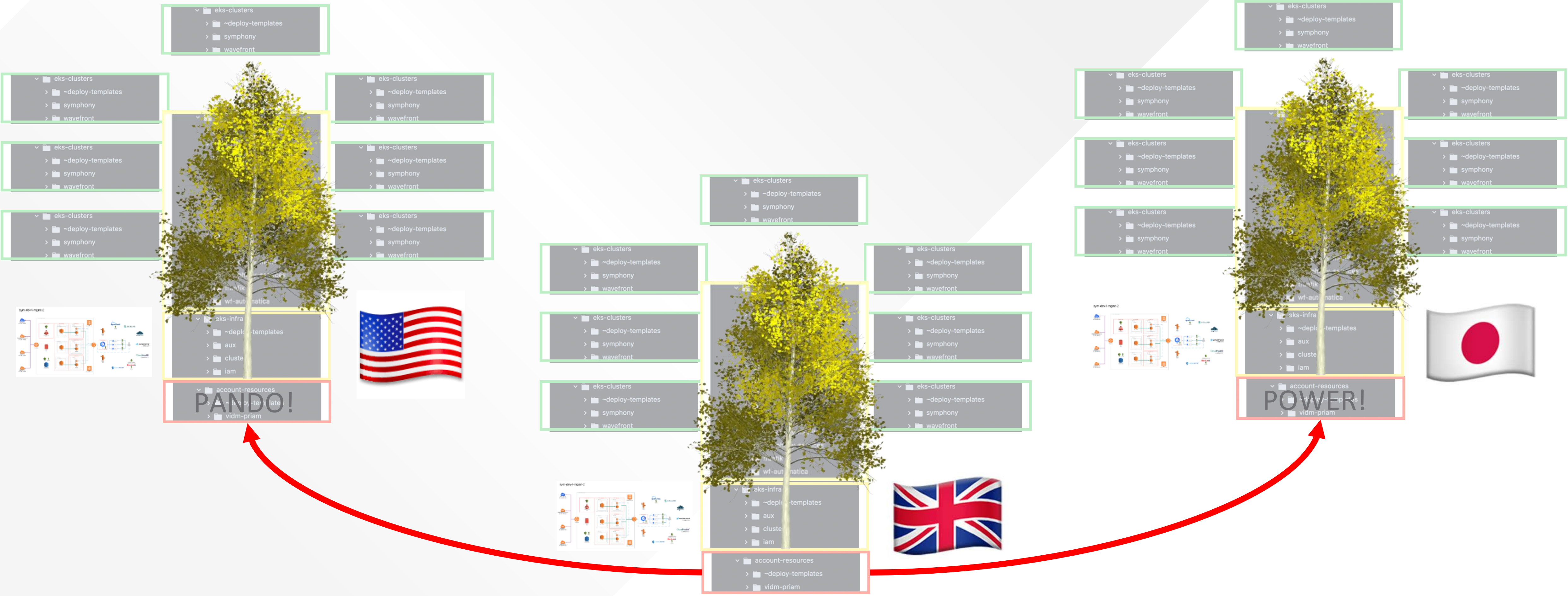
Level 1: Repeatable Environments



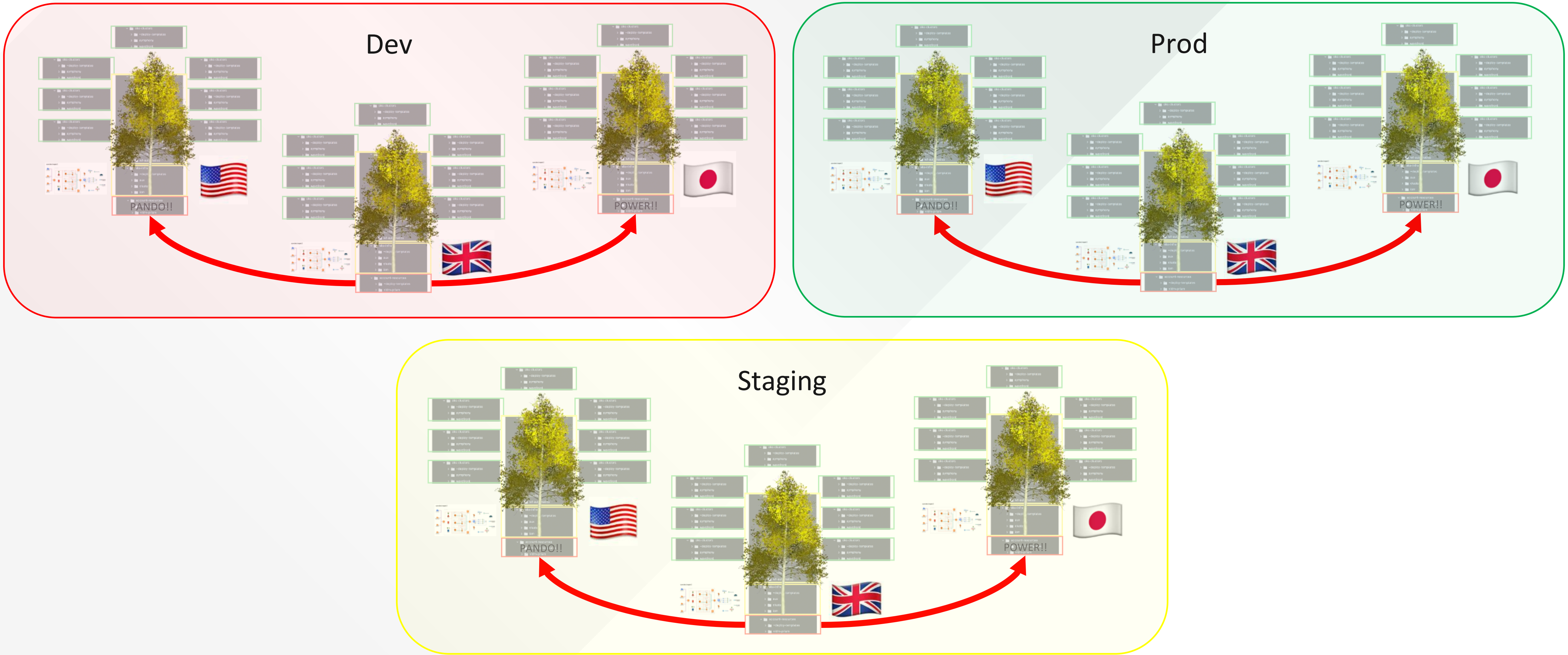
Level 1: Repeatable Environments



Level 2: Repeatable Environments Across Regions

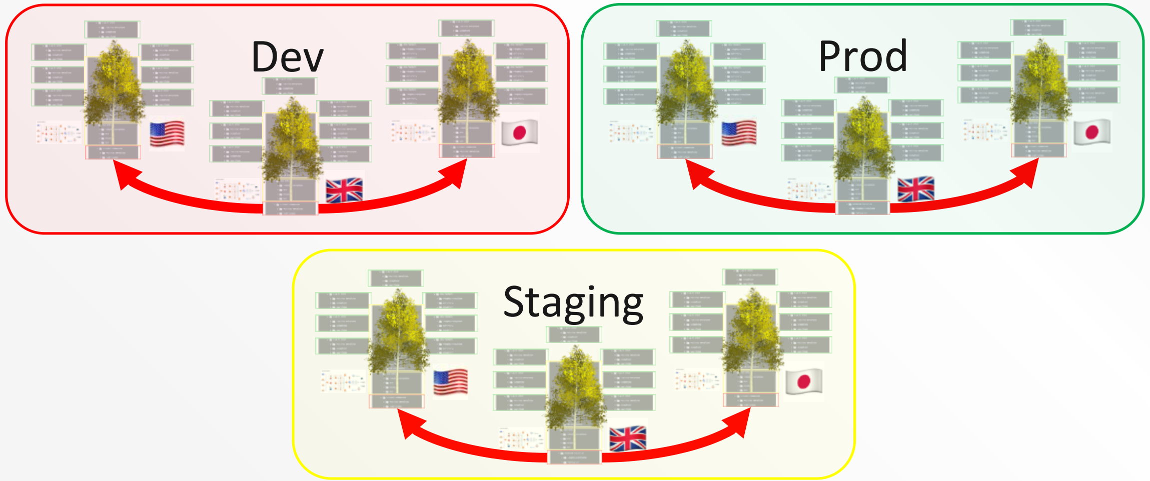


Level 3: Repeatable Environments Across Accounts

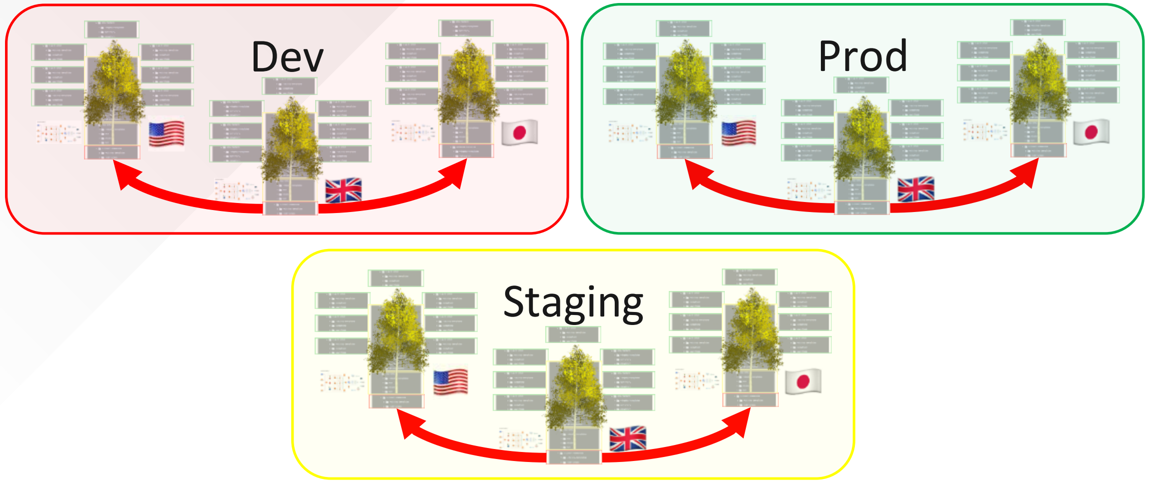


Level 4: Repeatable Environments Across Customers

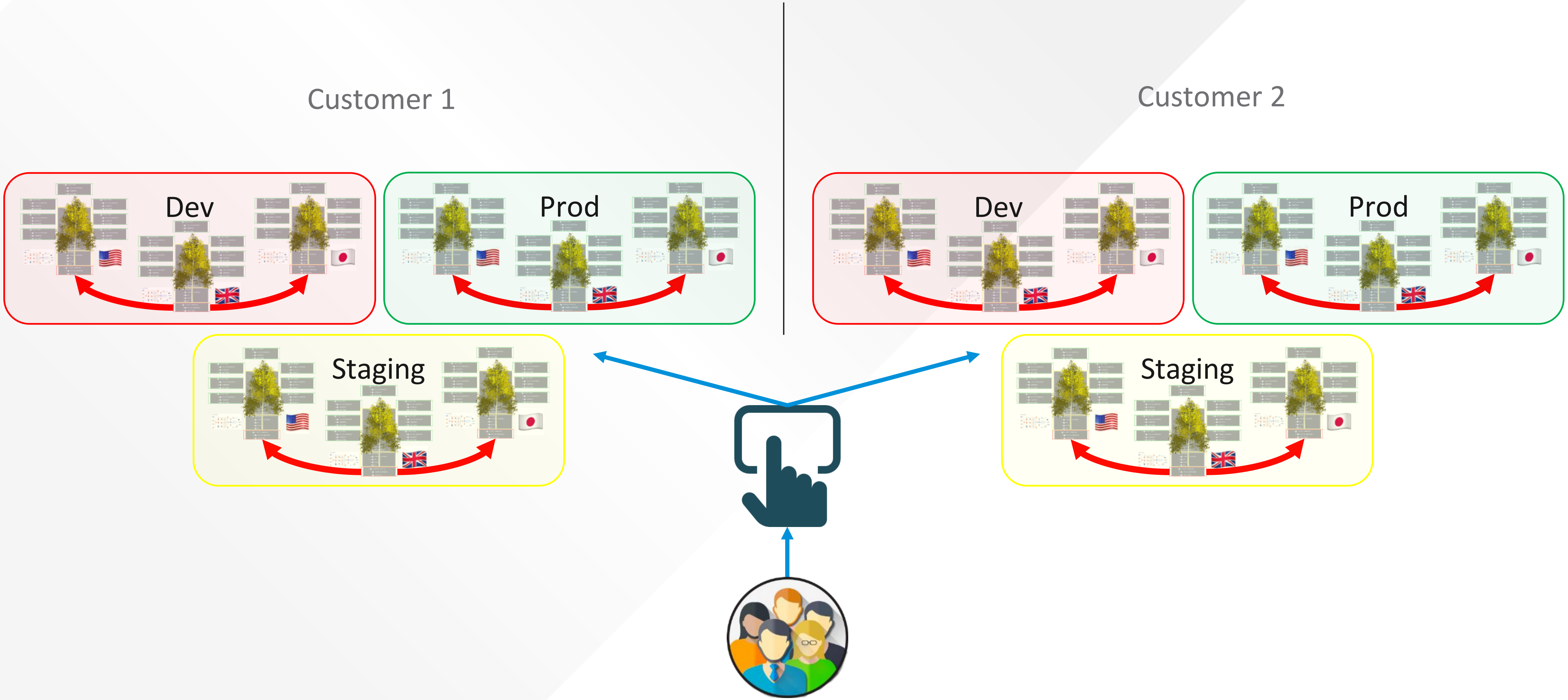
WAVEFRONT
by vmware



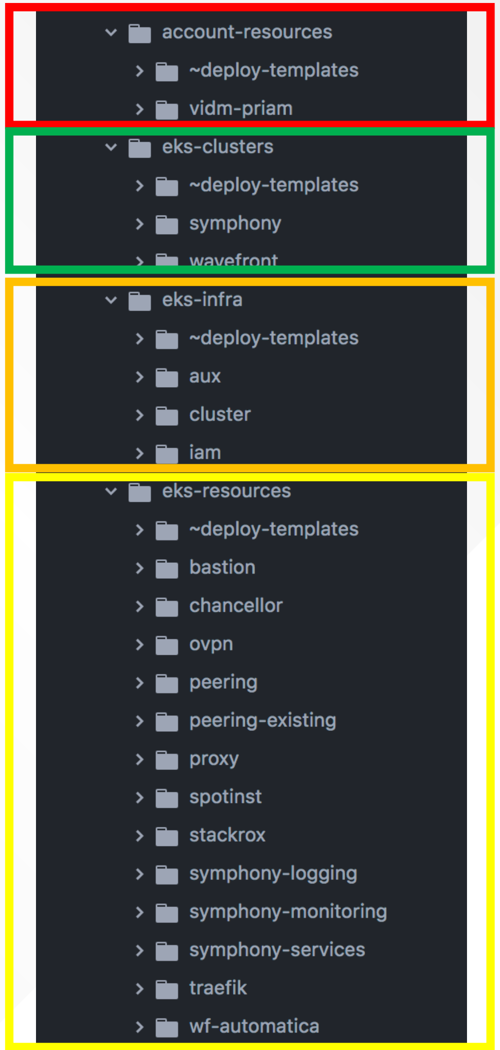
Symphony



Level 5: Self-Serve Environments



Code Structure



```
⌄ account-resources
    › ~deploy-templates
    › vidm-priam
⌄ eks-clusters
    › ~deploy-templates
    › symphony
    › wavefront
⌄ eks-infra
    › ~deploy-templates
    › aux
    › cluster
    › iam
⌄ eks-resources
    › ~deploy-templates
    › bastion
    › chancellor
    › ovpn
    › peering
    › peering-existing
    › proxy
    › spotinst
    › stackrox
    › symphony-logging
    › symphony-monitoring
    › symphony-services
    › traefik
    › wf-automatica
```

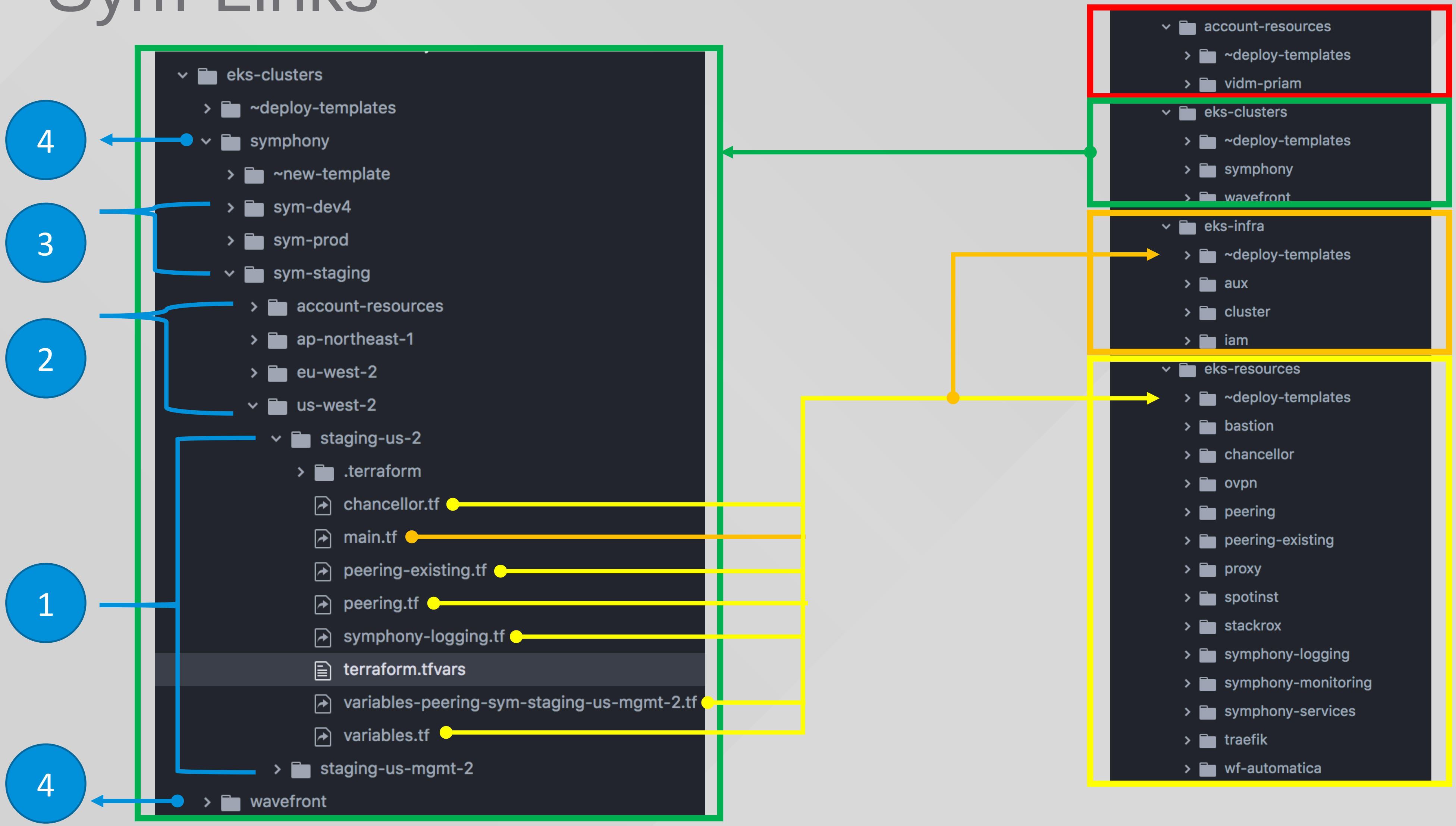
Run once per account
(required)

Cluster definitions
(required)

Cluster scaffolding
(required)

Cluster extensions
(optional)

Sym-Links



```
eks-clusters
  ~deploy-templates
  symphony
    ~new-template
    sym-dev4
    sym-prod
    sym-staging
      account-resources
      ap-northeast-1
      eu-west-2
      us-west-2
        staging-us-2
          .terraform
          chancellor.tf
          main.tf
          peering-existing.tf
          peering.tf
          symphony-logging.tf
          terraform.tfvars
          variables-peering-sym-staging-us-mgmt-2.tf
          variables.tf
        staging-us-mgmt-2
  wavefront
```

```
account-resources
  ~deploy-templates
  vidm-priam
eks-clusters
  ~deploy-templates
  symphony
  wavefront
eks-infra
  ~deploy-templates
  aux
  cluster
  iam
eks-resources
  ~deploy-templates
  bastion
  chancellor
  ovpn
  peering
  peering-existing
  proxy
  spotinst
  stackrox
  symphony-logging
  symphony-monitoring
  symphony-services
  traefik
  wf-automatica
```

Run once per account (required)

Cluster definitions (required)

Cluster scaffolding (required)

Cluster extensions (optional)

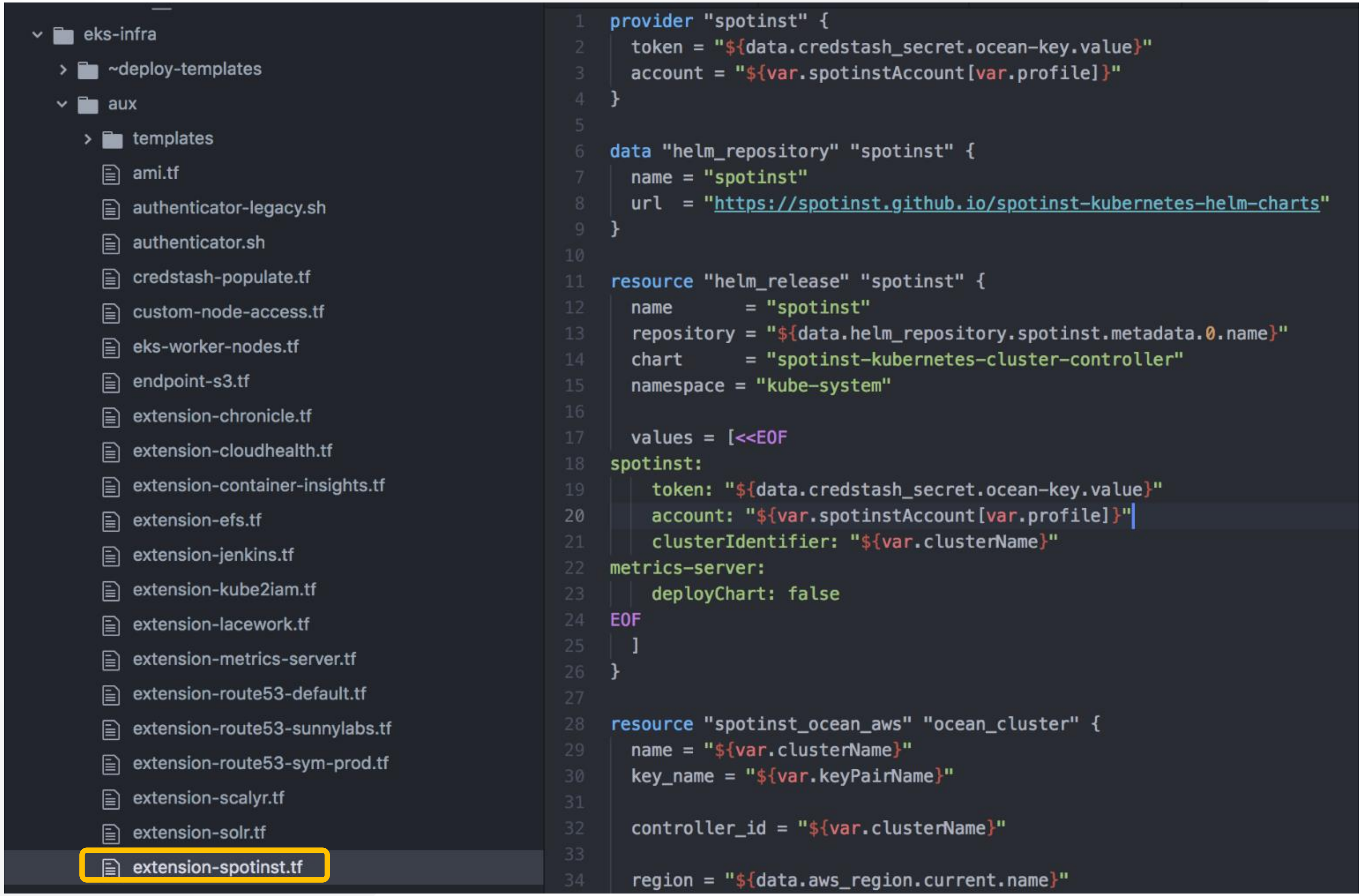
New Cluster Configurations

```
mkdir <new_cluster>
cp –PR ./<source_cluster> <new_cluster>

-P is to preserve sym-links
-R allow recursive directories
```

New Cluster Build

```
Terragrunt apply –target=module.iam (required eks-infra)
Terragrunt apply –target=module.cluster (required eks-infra)
Terragrunt apply –target=module.aux (required eks-infra)
Terragrunt apply –target=module.peering (optional)
Terragrunt apply –target=module.ovpn (optional)
Terragrunt apply –target=module.bastion (optional)
Terragrunt apply –target=module.proxy (optional)
```

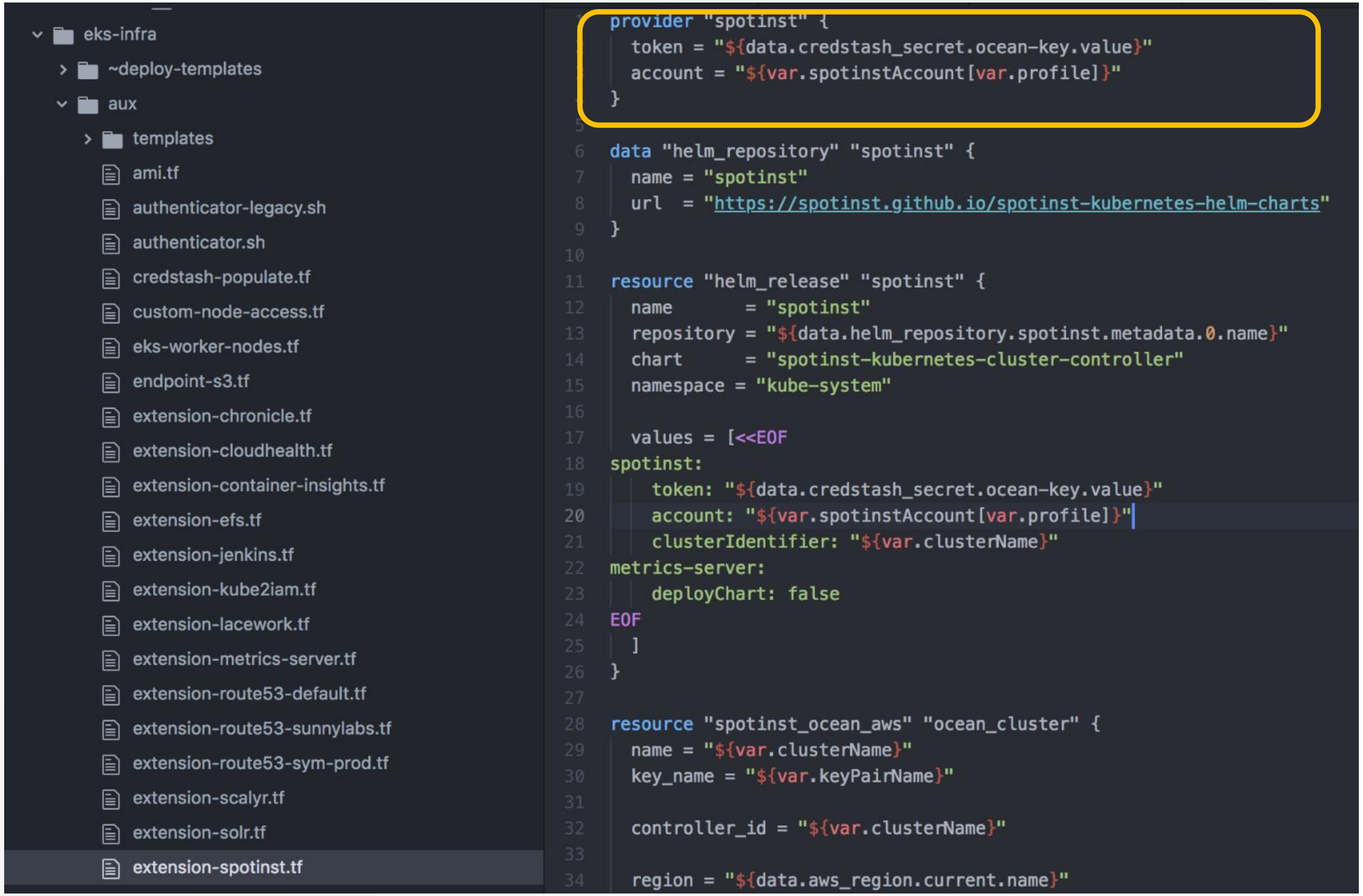
Spotinst Integration



```
eks-infra
  ~deploy-templates
  aux
    templates
    ami.tf
    authenticator-legacy.sh
    authenticator.sh
    credstash-populate.tf
    custom-node-access.tf
    eks-worker-nodes.tf
    endpoint-s3.tf
    extension-chronicle.tf
    extension-cloudhealth.tf
    extension-container-insights.tf
    extension-efs.tf
    extension-jenkins.tf
    extension-kube2iam.tf
    extension-lacework.tf
    extension-metrics-server.tf
    extension-route53-default.tf
    extension-route53-sunnylabs.tf
    extension-route53-sym-prod.tf
    extension-scalyr.tf
    extension-solr.tf
    extension-spotinst.tf
```

```
provider "spotinst" {
  token = "${data.credstash_secret.ocean-key.value}"
  account = "${var.spotinstAccount[var.profile]}"
}

data "helm_repository" "spotinst" {
  name = "spotinst"
  url  = "https://spotinst.github.io/spotinst-kubernetes-helm-charts"
}

resource "helm_release" "spotinst" {
  name       = "spotinst"
  repository = "${data.helm_repository.spotinst.metadata.0.name}"
  chart      = "spotinst-kubernetes-cluster-controller"
  namespace = "kube-system"

  values = [<<EOF
spotinst:
    token: "${data.credstash_secret.ocean-key.value}"
    account: "${var.spotinstAccount[var.profile]}"
    clusterIdentifier: "${var.clusterName}"
metrics-server:
    deployChart: false
EOF
  ]
}

resource "spotinst_ocean_aws" "ocean_cluster" {
  name = "${var.clusterName}"
  key_name = "${var.keyPairName}"

  controller_id = "${var.clusterName}"

  region = "${data.aws_region.current.name}"
```

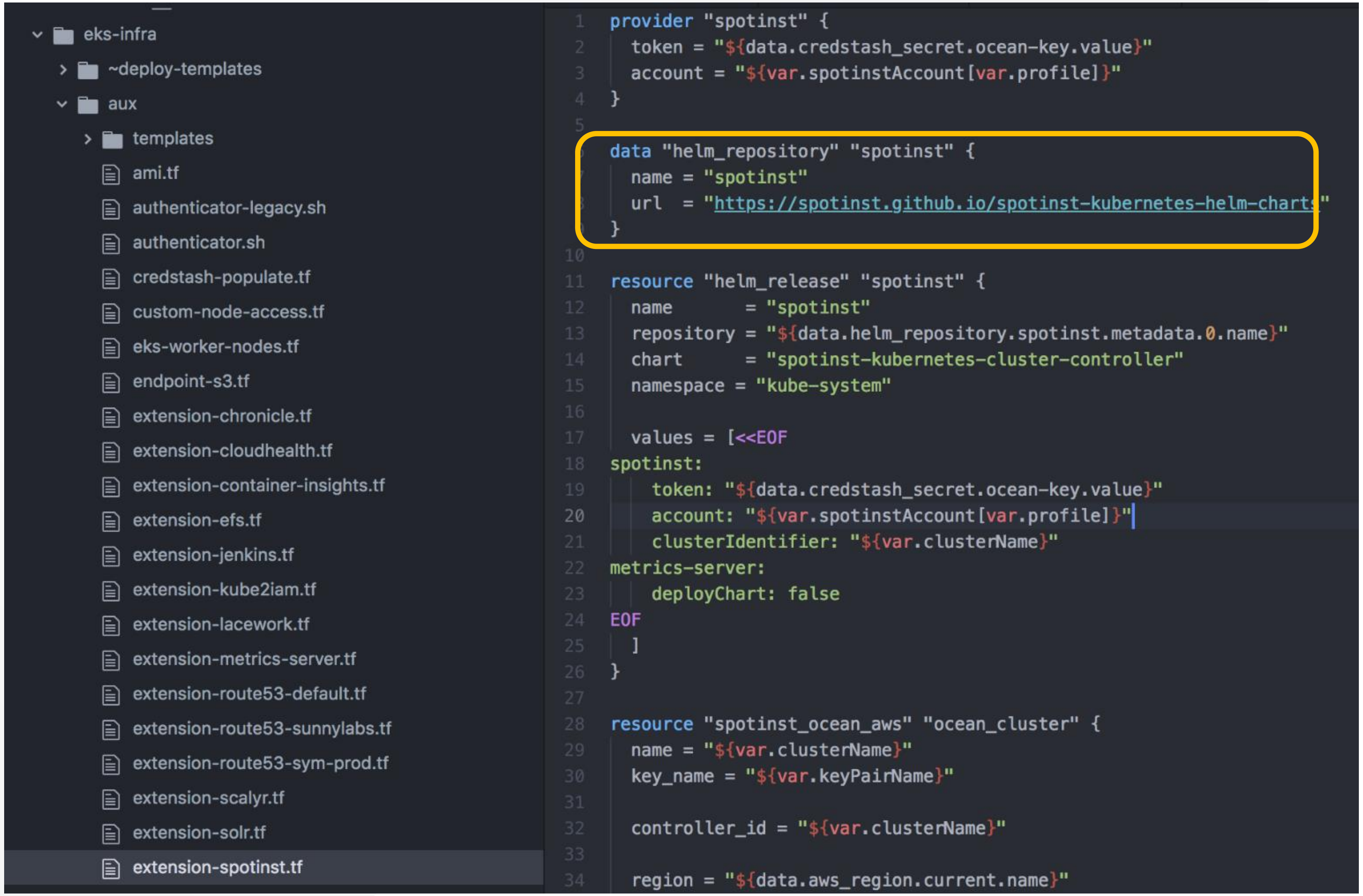
Spotinst Integration



- eks-infra
  - ~deploy-templates
  - aux
    - templates
    - ami.tf
    - authenticator-legacy.sh
    - authenticator.sh
    - credstash-populate.tf
    - custom-node-access.tf
    - eks-worker-nodes.tf
    - endpoint-s3.tf
    - extension-chronicle.tf
    - extension-cloudhealth.tf
    - extension-container-insights.tf
    - extension-efs.tf
    - extension-jenkins.tf
    - extension-kube2iam.tf
    - extension-lacework.tf
    - extension-metrics-server.tf
    - extension-route53-default.tf
    - extension-route53-sunnylabs.tf
    - extension-route53-sym-prod.tf
    - extension-scalyr.tf
    - extension-solr.tf
    - extension-spotinst.tf

```
provider "spotinst" {
  token = "${data.credstash_secret.ocean-key.value}"
  account = "${var.spotinstAccount[var.profile]}"
}

data "helm_repository" "spotinst" {
  name = "spotinst"
  url  = "https://spotinst.github.io/spotinst-kubernetes-helm-charts"
}

resource "helm_release" "spotinst" {
  name       = "spotinst"
  repository = "${data.helm_repository.spotinst.metadata.0.name}"
  chart      = "spotinst-kubernetes-cluster-controller"
  namespace = "kube-system"

  values = [<<EOF
spotinst:
    token: "${data.credstash_secret.ocean-key.value}"
    account: "${var.spotinstAccount[var.profile]}"
    clusterIdentifier: "${var.clusterName}"
metrics-server:
    deployChart: false
EOF
  ]
}

resource "spotinst_ocean_aws" "ocean_cluster" {
  name = "${var.clusterName}"
  key_name = "${var.keyPairName}"

  controller_id = "${var.clusterName}"

  region = "${data.aws_region.current.name}"
```

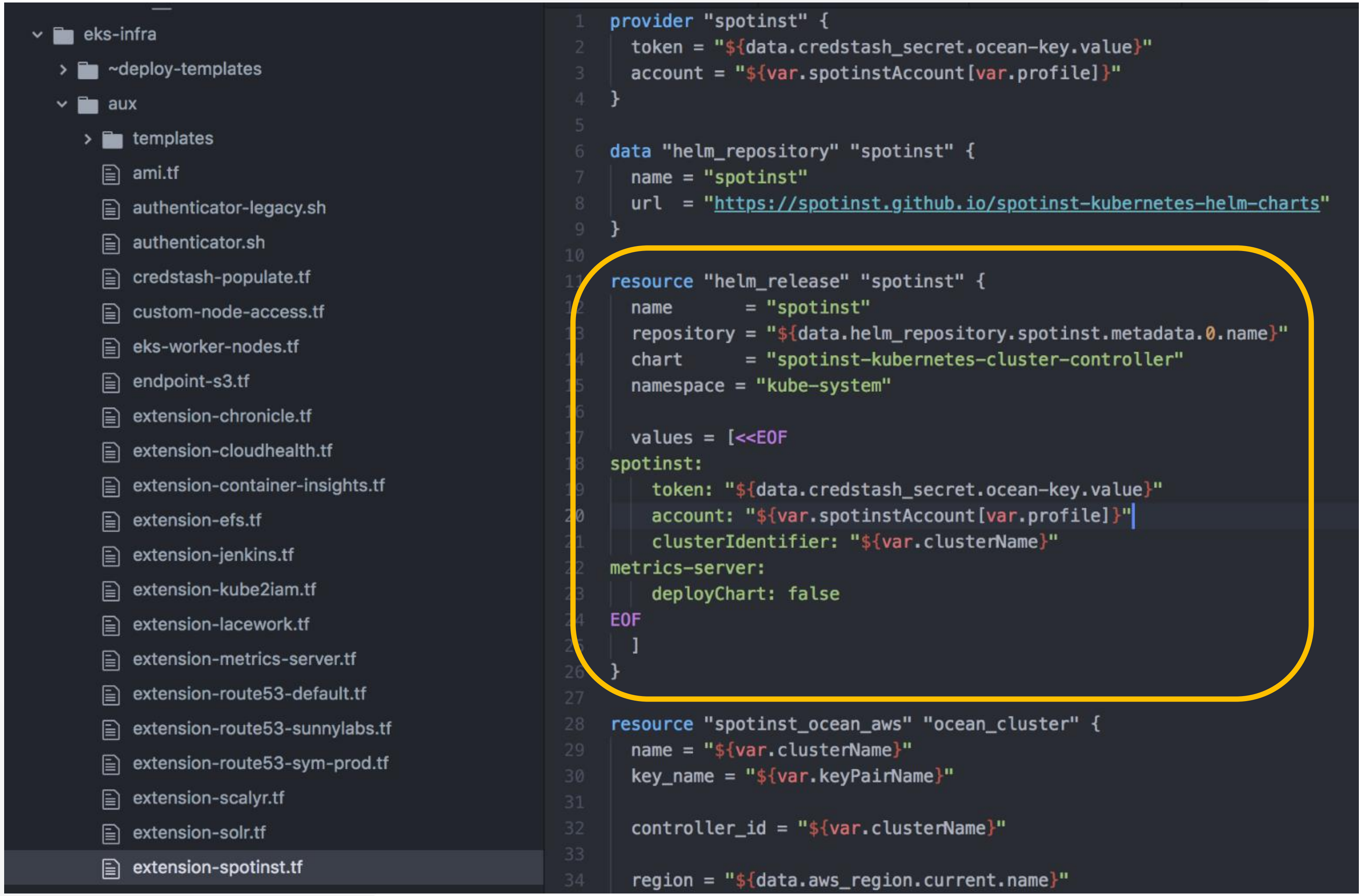
Spotinst Integration



- eks-infra
  - ~deploy-templates
  - aux
    - templates
    - ami.tf
    - authenticator-legacy.sh
    - authenticator.sh
    - credstash-populate.tf
    - custom-node-access.tf
    - eks-worker-nodes.tf
    - endpoint-s3.tf
    - extension-chronicle.tf
    - extension-cloudhealth.tf
    - extension-container-insights.tf
    - extension-efs.tf
    - extension-jenkins.tf
    - extension-kube2iam.tf
    - extension-lacework.tf
    - extension-metrics-server.tf
    - extension-route53-default.tf
    - extension-route53-sunnylabs.tf
    - extension-route53-sym-prod.tf
    - extension-scalyr.tf
    - extension-solr.tf
    - extension-spotinst.tf

```
provider "spotinst" {
  token = "${data.credstash_secret.ocean-key.value}"
  account = "${var.spotinstAccount[var.profile]}"
}

data "helm_repository" "spotinst" {
  name = "spotinst"
  url  = "https://spotinst.github.io/spotinst-kubernetes-helm-charts"
}

resource "helm_release" "spotinst" {
  name       = "spotinst"
  repository = "${data.helm_repository.spotinst.metadata.0.name}"
  chart      = "spotinst-kubernetes-cluster-controller"
  namespace = "kube-system"

  values = [<<EOF
spotinst:
    token: "${data.credstash_secret.ocean-key.value}"
    account: "${var.spotinstAccount[var.profile]}"
    clusterIdentifier: "${var.clusterName}"
metrics-server:
    deployChart: false
EOF
  ]
}

resource "spotinst_ocean_aws" "ocean_cluster" {
  name = "${var.clusterName}"
  key_name = "${var.keyPairName}"

  controller_id = "${var.clusterName}"

  region = "${data.aws_region.current.name}"
```

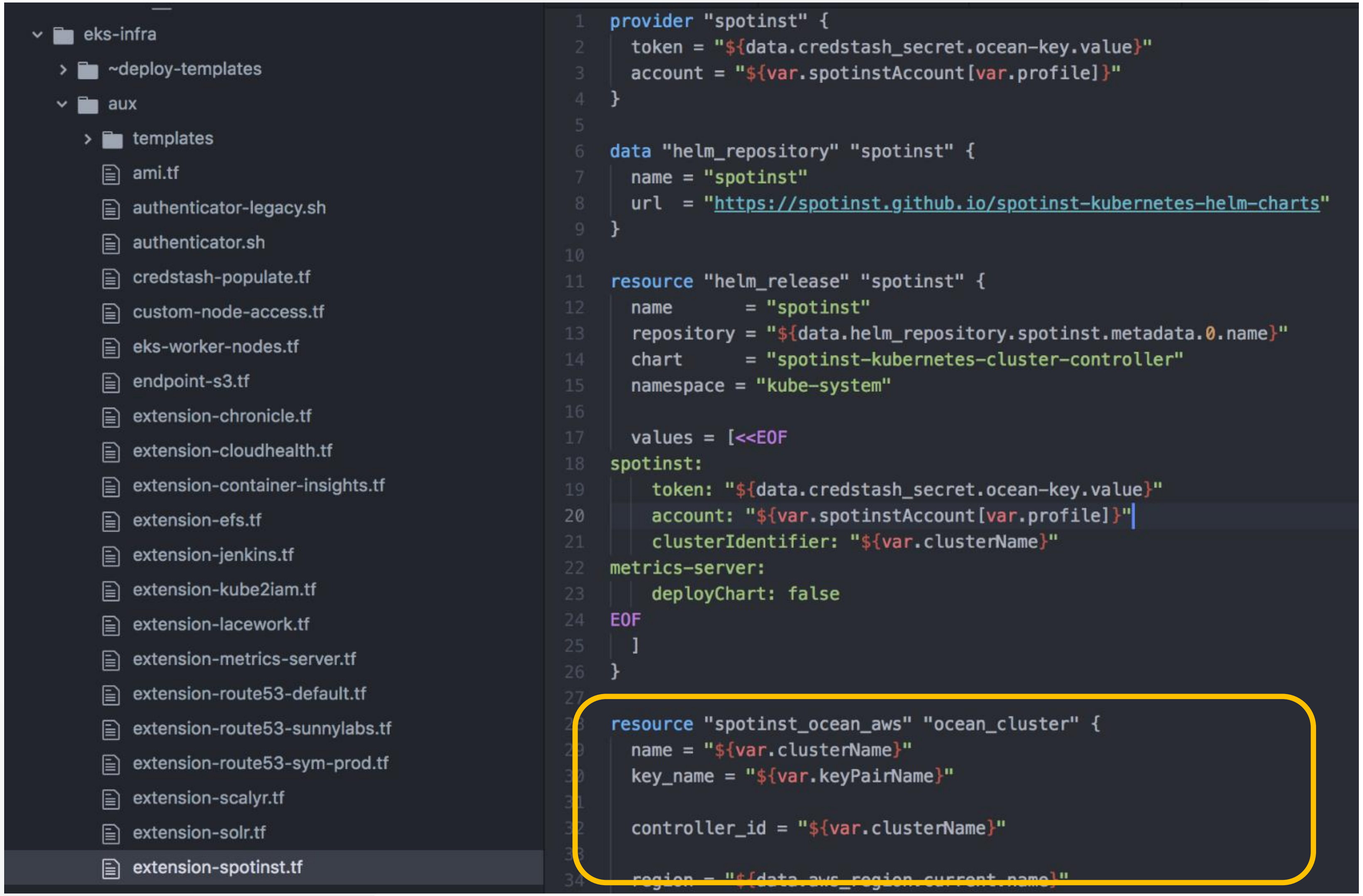
Spotinst Integration



- eks-infra
  - ~deploy-templates
  - aux
    - templates
    - ami.tf
    - authenticator-legacy.sh
    - authenticator.sh
    - credstash-populate.tf
    - custom-node-access.tf
    - eks-worker-nodes.tf
    - endpoint-s3.tf
    - extension-chronicle.tf
    - extension-cloudhealth.tf
    - extension-container-insights.tf
    - extension-efs.tf
    - extension-jenkins.tf
    - extension-kube2iam.tf
    - extension-lacework.tf
    - extension-metrics-server.tf
    - extension-route53-default.tf
    - extension-route53-sunnylabs.tf
    - extension-route53-sym-prod.tf
    - extension-scalyr.tf
    - extension-solr.tf
    - extension-spotinst.tf

```
provider "spotinst" {
  token = "${data.credstash_secret.ocean-key.value}"
  account = "${var.spotinstAccount[var.profile]}"
}

data "helm_repository" "spotinst" {
  name = "spotinst"
  url  = "https://spotinst.github.io/spotinst-kubernetes-helm-charts"
}

resource "helm_release" "spotinst" {
  name       = "spotinst"
  repository = "${data.helm_repository.spotinst.metadata.0.name}"
  chart      = "spotinst-kubernetes-cluster-controller"
  namespace = "kube-system"

  values = [<<EOF
spotinst:
  token: "${data.credstash_secret.ocean-key.value}"
  account: "${var.spotinstAccount[var.profile]}"
  clusterIdentifier: "${var.clusterName}"
metrics-server:
  deployChart: false
EOF
  ]
}

resource "spotinst_ocean_aws" "ocean_cluster" {
  name = "${var.clusterName}"
  key_name = "${var.keyPairName}"

  controller_id = "${var.clusterName}"

  region = "${data.aws_region.current.name}"
```

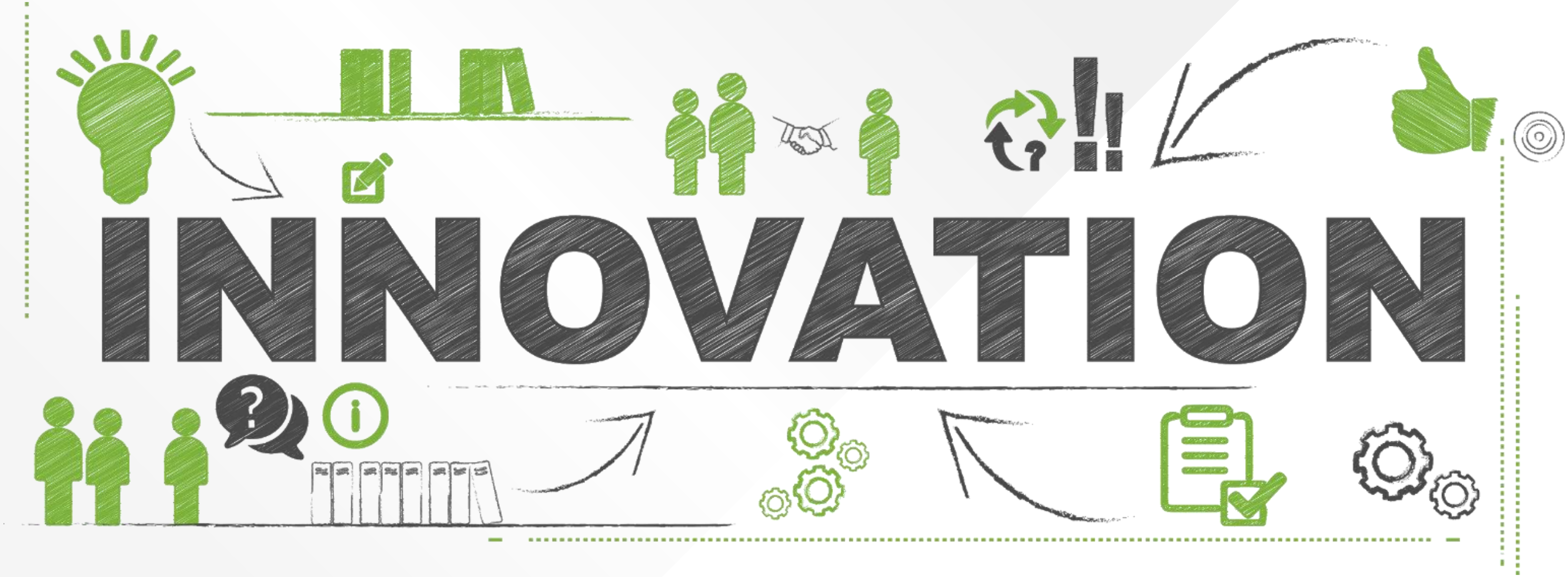
Spotinst Integration



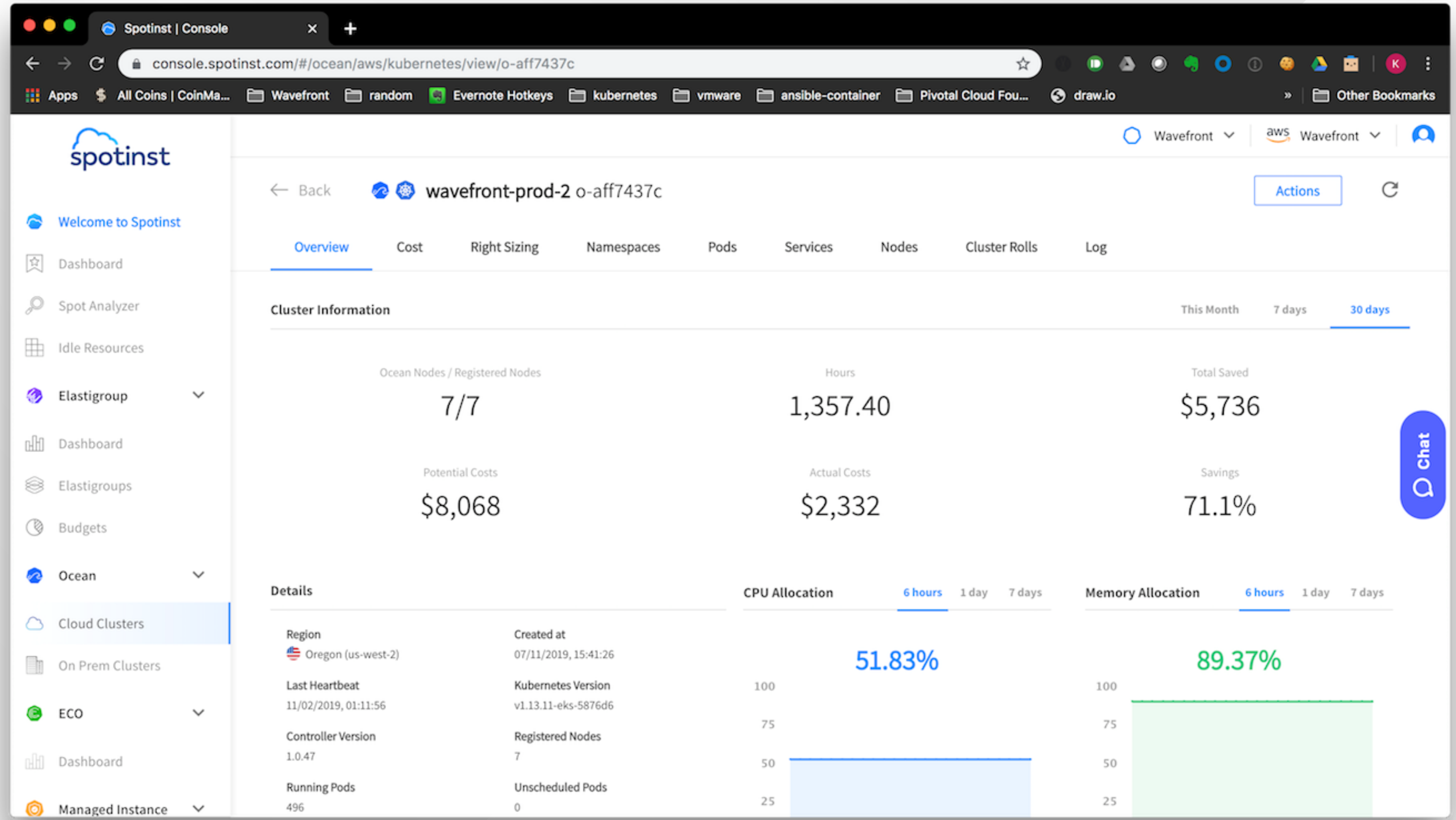
- eks-infra
  - ~deploy-templates
  - aux
    - templates
    - ami.tf
    - authenticator-legacy.sh
    - authenticator.sh
    - credstash-populate.tf
    - custom-node-access.tf
    - eks-worker-nodes.tf
    - endpoint-s3.tf
    - extension-chronicle.tf
    - extension-cloudhealth.tf
    - extension-container-insights.tf
    - extension-efs.tf
    - extension-jenkins.tf
    - extension-kube2iam.tf
    - extension-lacework.tf
    - extension-metrics-server.tf
    - extension-route53-default.tf
    - extension-route53-sunnylabs.tf
    - extension-route53-sym-prod.tf
    - extension-scalyr.tf
    - extension-solr.tf
    - extension-spotinst.tf

```
provider "spotinst" {
  token = "${data.credstash_secret.ocean-key.value}"
  account = "${var.spotinstAccount[var.profile]}"
}

data "helm_repository" "spotinst" {
  name = "spotinst"
  url  = "https://spotinst.github.io/spotinst-kubernetes-helm-charts"
}

resource "helm_release" "spotinst" {
  name       = "spotinst"
  repository = "${data.helm_repository.spotinst.metadata.0.name}"
  chart      = "spotinst-kubernetes-cluster-controller"
  namespace = "kube-system"

  values = [<<EOF
spotinst:
    token: "${data.credstash_secret.ocean-key.value}"
    account: "${var.spotinstAccount[var.profile]}"
    clusterIdentifier: "${var.clusterName}"
metrics-server:
    deployChart: false
EOF
  ]
}

resource "spotinst_ocean_aws" "ocean_cluster" {
  name = "${var.clusterName}"
  key_name = "${var.keyPairName}"

  controller_id = "${var.clusterName}"

  region = "${data.aws_region.current.name}"
```

Price
Value


INNOVATION

30-Day Savings on a Single Cluster



wavefront-prod-2 o-aff7437c

Cluster Information

Ocean Nodes / Registered Nodes	Hours	Total Saved
7/7	1,357.40	$5,736
Potential Costs	**Actual Costs**	**Savings**
$8,068	$2,332	71.1%

Details

Region
Oregon (us-west-2)

Created at
07/11/2019, 15:41:26

Last Heartbeat
11/02/2019, 01:11:56

Kubernetes Version
v1.13.11-eks-5876d6

Controller Version
1.0.47

Registered Nodes
7

Running Pods
496

Unscheduled Pods
0

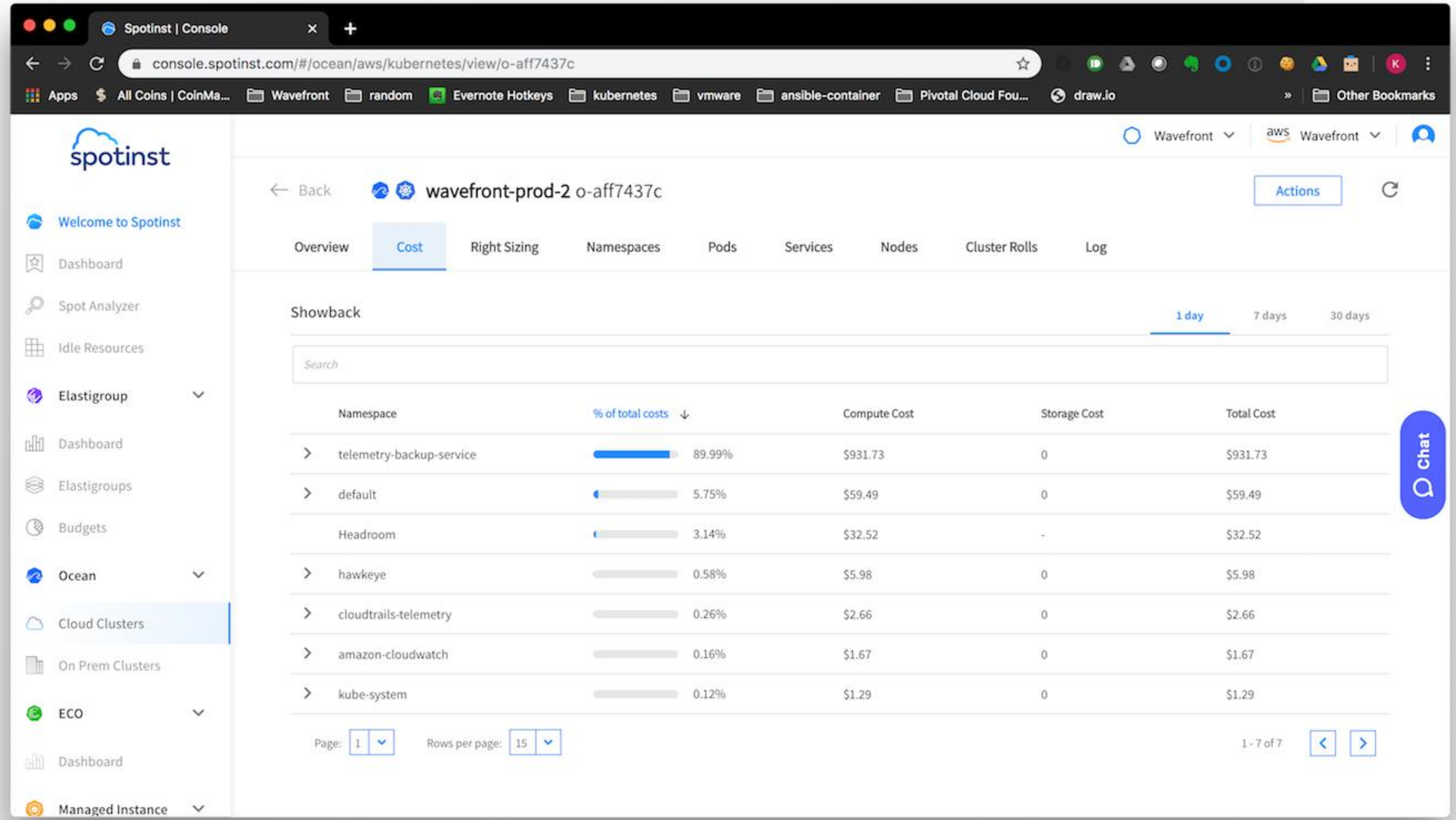
CPU Allocation

51.83%

Memory Allocation

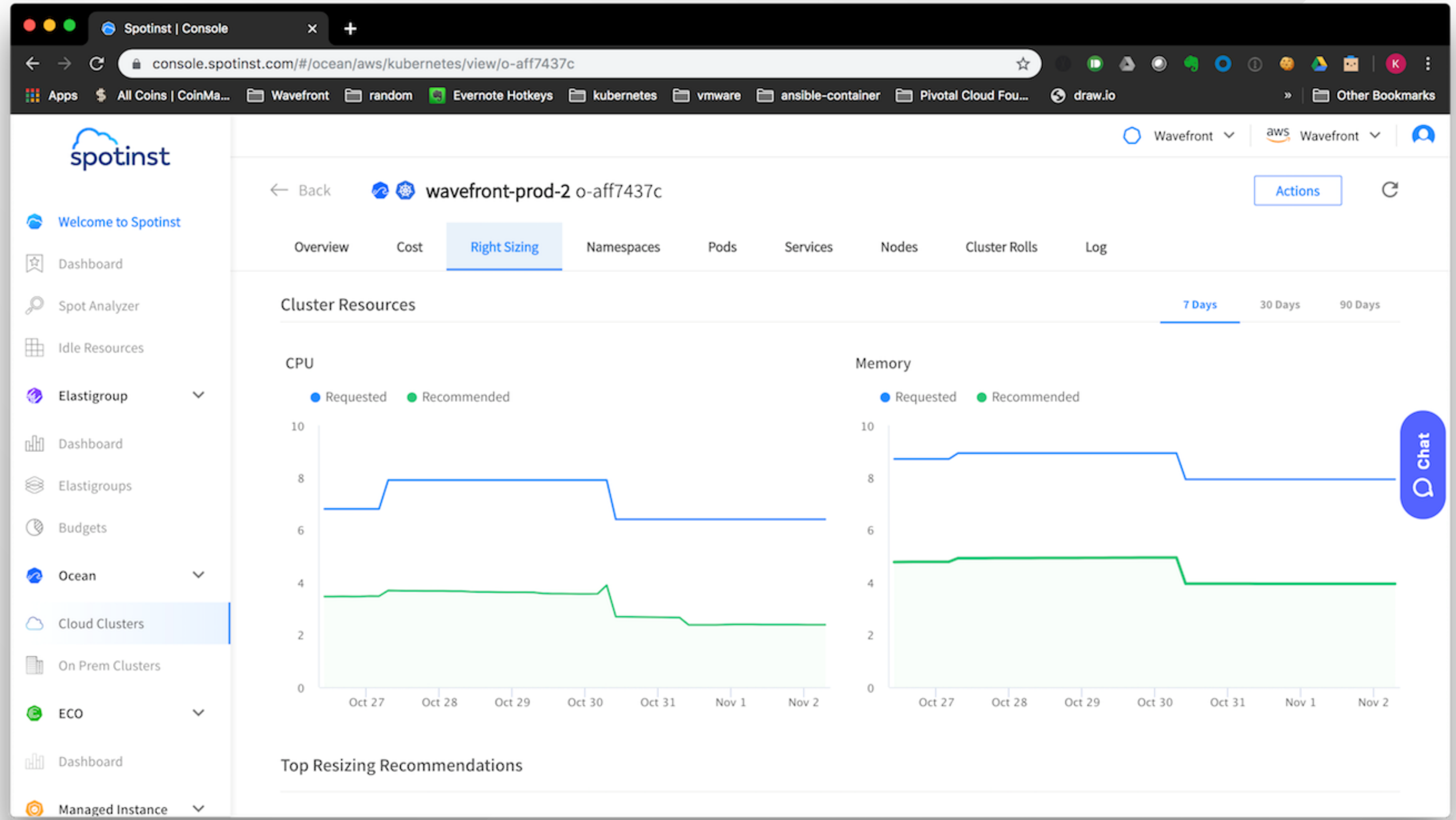
89.37%

Cost by Namespace

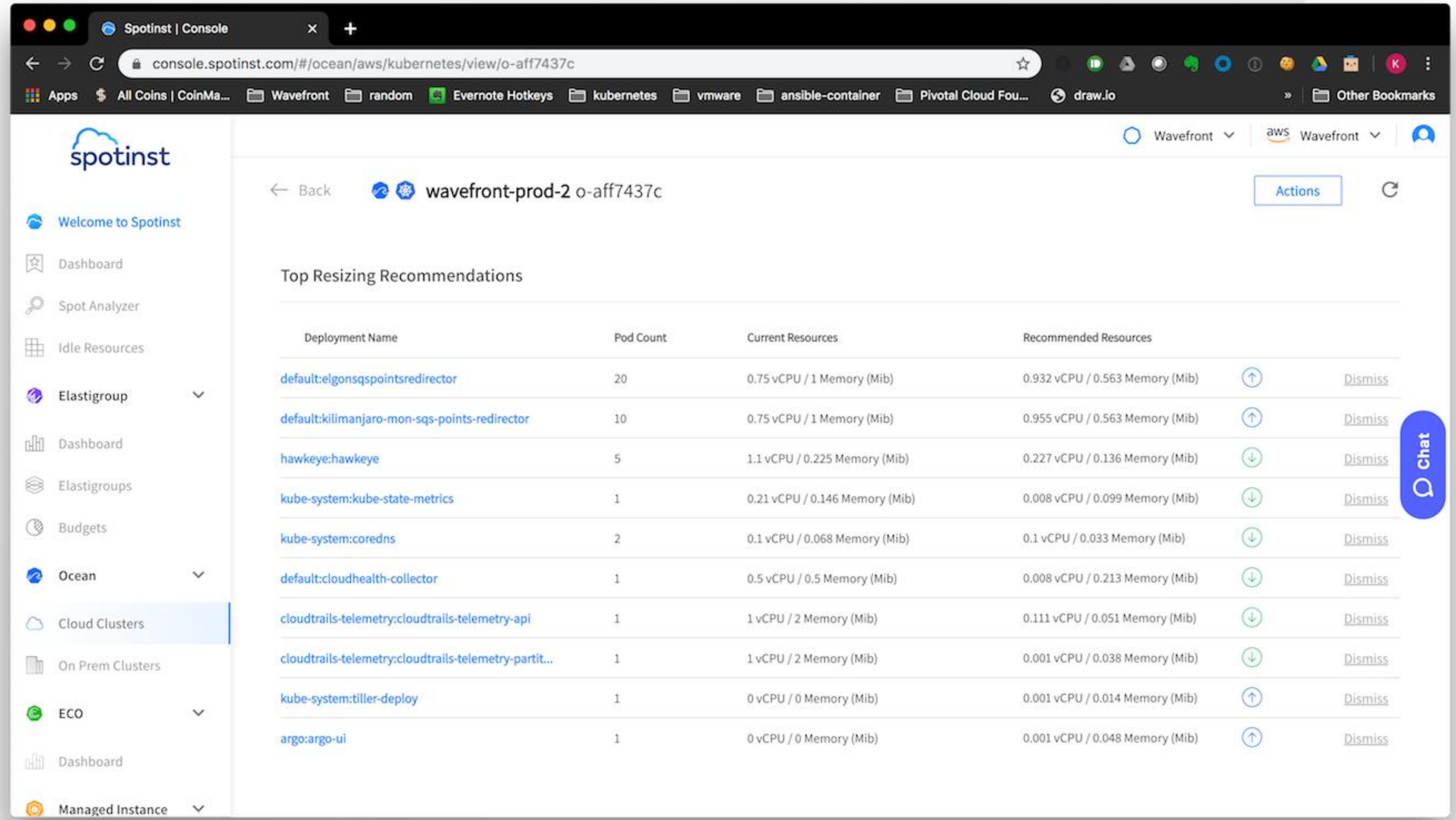


Namespace	% of total costs ↓	Compute Cost	Storage Cost	Total Cost
telemetry-backup-service	89.99%	$931.73	0	$931.73
default	5.75%	$59.49	0	$59.49
Headroom	3.14%	$32.52	-	$32.52
hawkeye	0.58%	$5.98	0	$5.98
cloudtrails-telemetry	0.26%	$2.66	0	$2.66
amazon-cloudwatch	0.16%	$1.67	0	$1.67
kube-system	0.12%	$1.29	0	$1.29

Workload Rightsizing



Workload Rightsizing

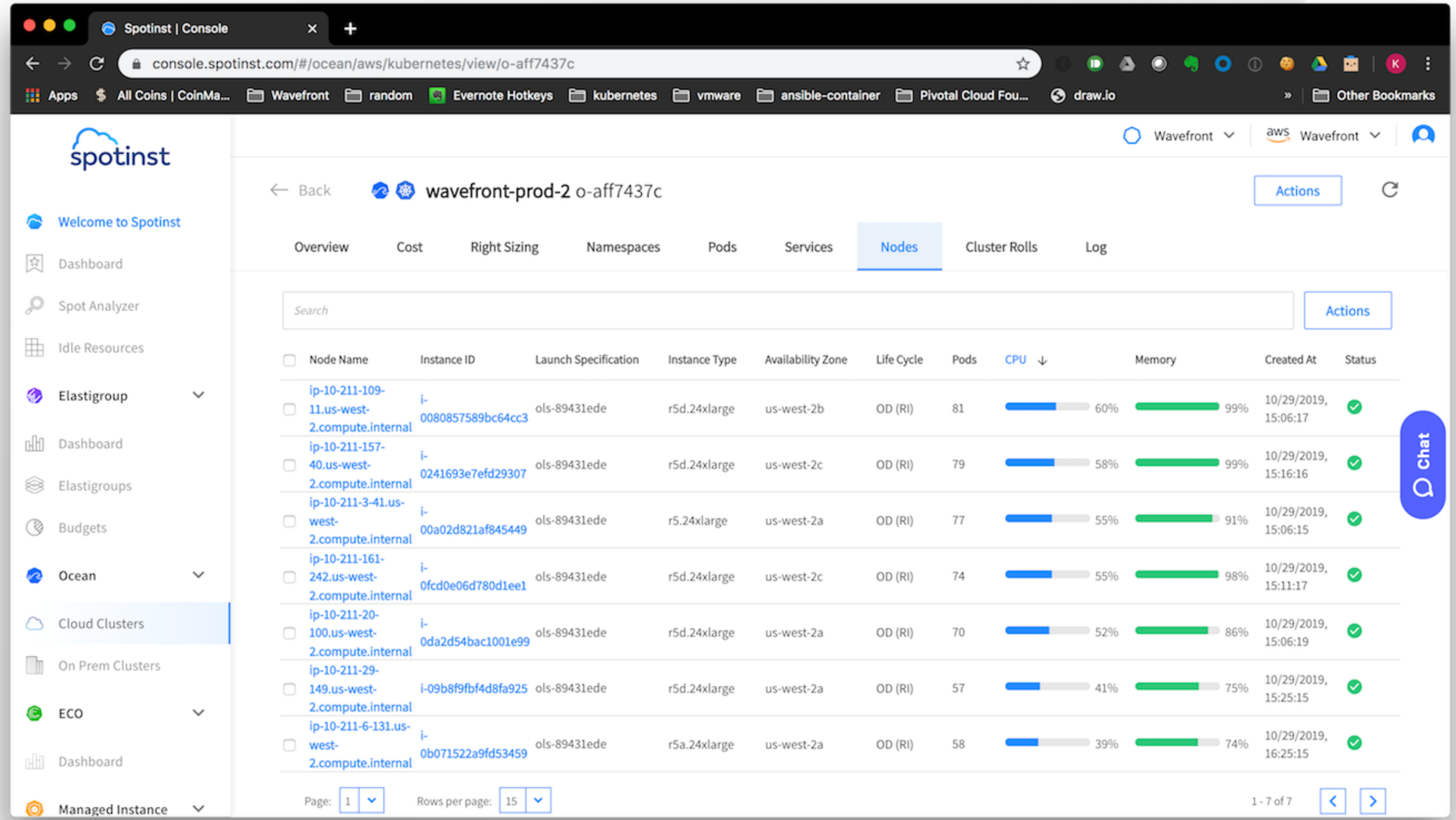


wavefront-prod-2 o-aff7437c

Top Resizing Recommendations

Deployment Name	Pod Count	Current Resources	Recommended Resources		
default:elgonsqspointsredirector	20	0.75 vCPU / 1 Memory (Mib)	0.932 vCPU / 0.563 Memory (Mib)	↑	Dismiss
default:kilimanjaro-mon-sqs-points-redirector	10	0.75 vCPU / 1 Memory (Mib)	0.955 vCPU / 0.563 Memory (Mib)	↑	Dismiss
hawkeye:hawkeye	5	1.1 vCPU / 0.225 Memory (Mib)	0.227 vCPU / 0.136 Memory (Mib)	↓	Dismiss
kube-system:kube-state-metrics	1	0.21 vCPU / 0.146 Memory (Mib)	0.008 vCPU / 0.099 Memory (Mib)	↓	Dismiss
kube-system:coredns	2	0.1 vCPU / 0.068 Memory (Mib)	0.1 vCPU / 0.033 Memory (Mib)	↓	Dismiss
default:cloudhealth-collector	1	0.5 vCPU / 0.5 Memory (Mib)	0.008 vCPU / 0.213 Memory (Mib)	↓	Dismiss
cloudtrails-telemetry:cloudtrails-telemetry-api	1	1 vCPU / 2 Memory (Mib)	0.111 vCPU / 0.051 Memory (Mib)	↓	Dismiss
cloudtrails-telemetry:cloudtrails-telemetry-partit...	1	1 vCPU / 2 Memory (Mib)	0.001 vCPU / 0.038 Memory (Mib)	↓	Dismiss
kube-system:tiller-deploy	1	0 vCPU / 0 Memory (Mib)	0.001 vCPU / 0.014 Memory (Mib)	↑	Dismiss
argo:argo-ui	1	0 vCPU / 0 Memory (Mib)	0.001 vCPU / 0.048 Memory (Mib)	↑	Dismiss

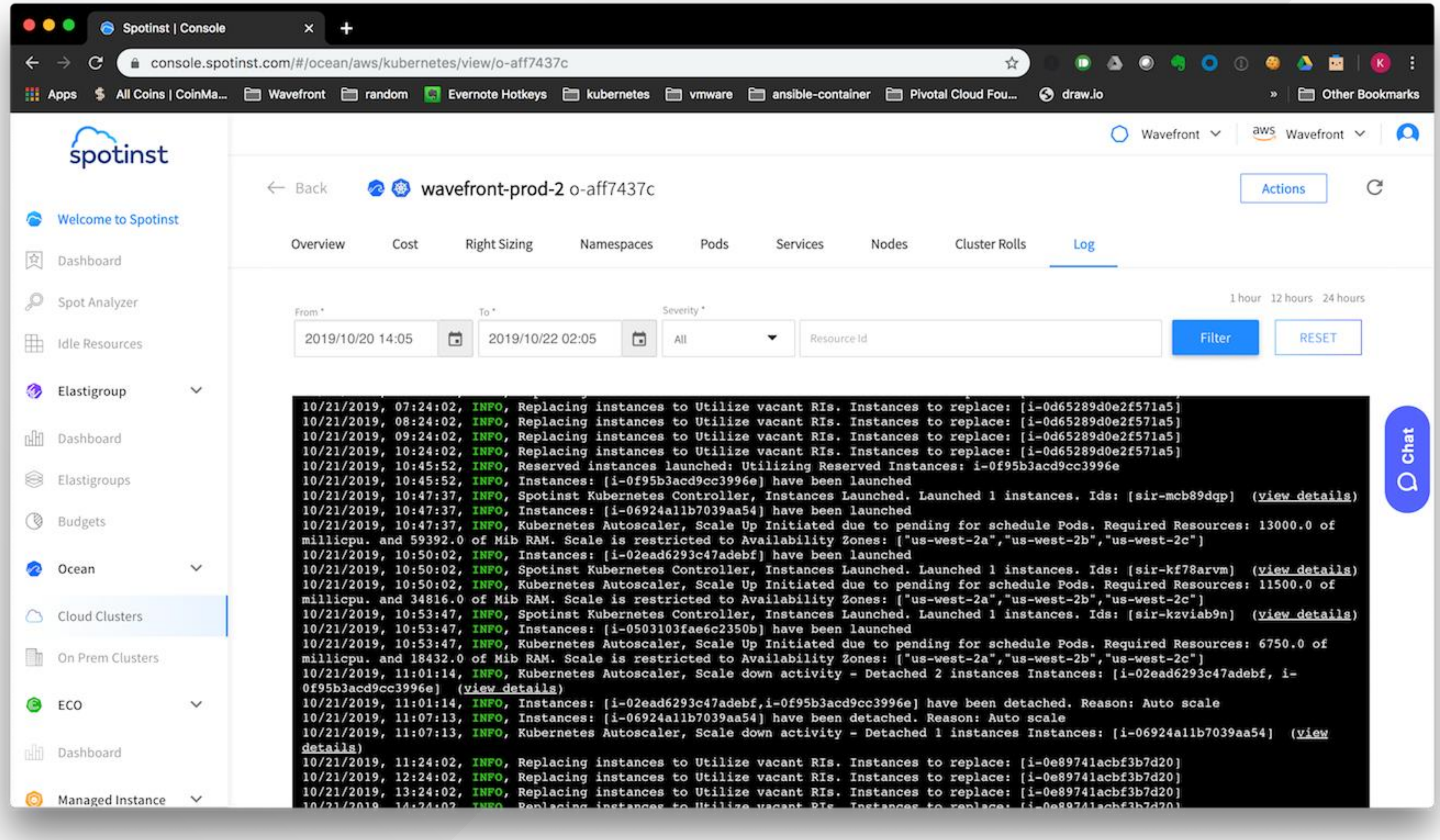
Node Statistics



□	Node Name	Instance ID	Launch Specification	Instance Type	Availability Zone	Life Cycle	Pods	CPU ↓	Memory	Created At	Status
□	ip-10-211-109-11.us-west-2.compute.internal	i-0080857589bc64cc3	ols-89431ede	r5d.24xlarge	us-west-2b	OD (RI)	81	60%	99%	10/29/2019, 15:06:17	✓
□	ip-10-211-157-40.us-west-2.compute.internal	i-0241693e7efd29307	ols-89431ede	r5d.24xlarge	us-west-2c	OD (RI)	79	58%	99%	10/29/2019, 15:16:16	✓
□	ip-10-211-3-41.us-west-2.compute.internal	i-00a02d821af845449	ols-89431ede	r5.24xlarge	us-west-2a	OD (RI)	77	55%	91%	10/29/2019, 15:06:15	✓
□	ip-10-211-161-242.us-west-2.compute.internal	i-0fcd0e06d780d1ee1	ols-89431ede	r5d.24xlarge	us-west-2c	OD (RI)	74	55%	98%	10/29/2019, 15:11:17	✓
□	ip-10-211-20-100.us-west-2.compute.internal	i-0da2d54bac1001e99	ols-89431ede	r5d.24xlarge	us-west-2a	OD (RI)	70	52%	86%	10/29/2019, 15:06:19	✓
□	ip-10-211-29-149.us-west-2.compute.internal	i-09b8f9fbf4d8fa925	ols-89431ede	r5d.24xlarge	us-west-2a	OD (RI)	57	41%	75%	10/29/2019, 15:25:15	✓
□	ip-10-211-6-131.us-west-2.compute.internal	i-0b071522a9fd53459	ols-89431ede	r5a.24xlarge	us-west-2a	OD (RI)	58	39%	74%	10/29/2019, 16:25:15	✓

Page: 1 Rows per page: 15 1 - 7 of 7

Scale Behavior – Spotinst UI



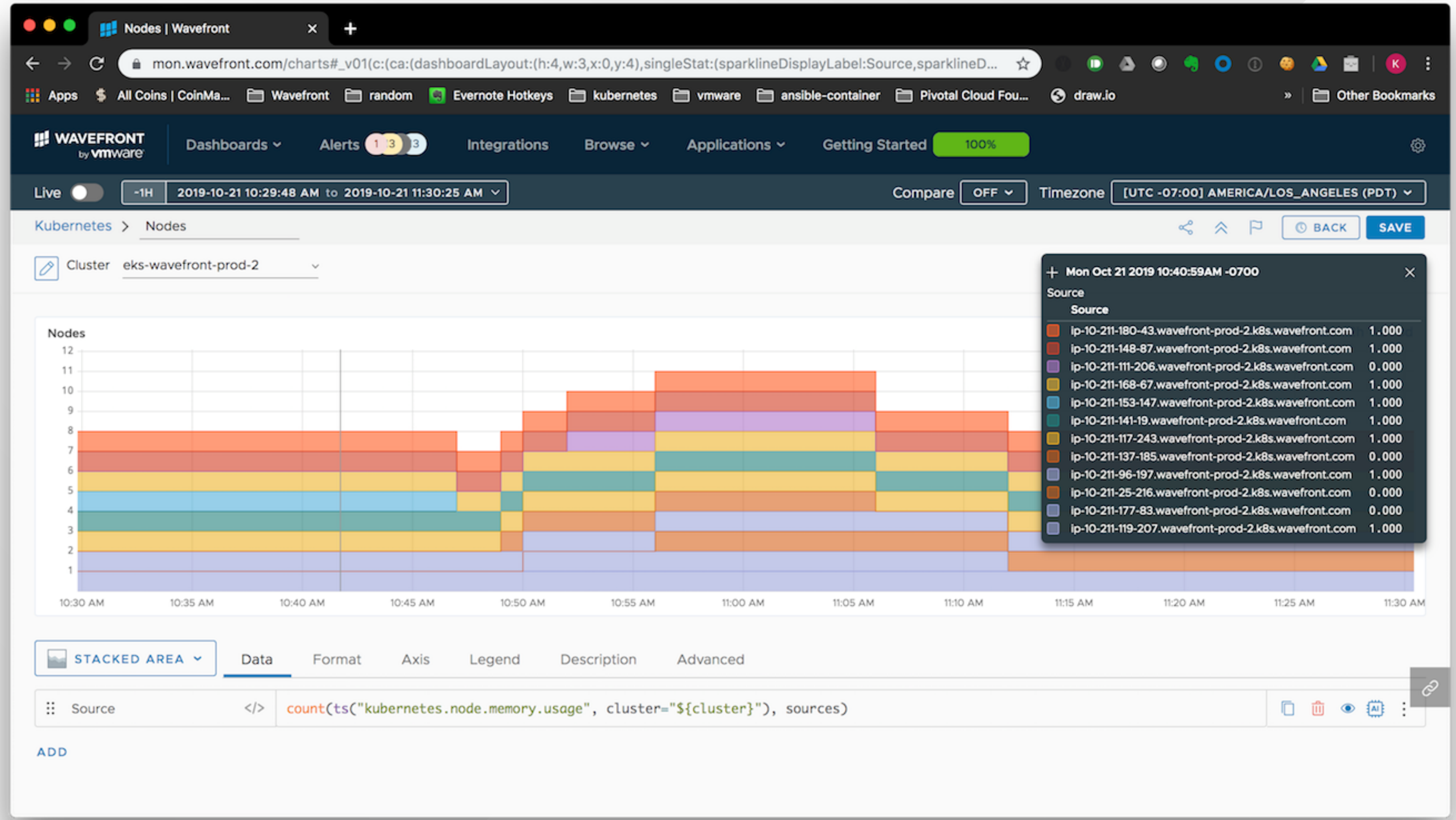
From * 2019/10/20 14:05

To * 2019/10/22 02:05

Severity * All

```
10/21/2019, 07:24:02, INFO, Replacing instances to Utilize vacant RIs. Instances to replace: [i-0d65289d0e2f571a5]
10/21/2019, 08:24:02, INFO, Replacing instances to Utilize vacant RIs. Instances to replace: [i-0d65289d0e2f571a5]
10/21/2019, 09:24:02, INFO, Replacing instances to Utilize vacant RIs. Instances to replace: [i-0d65289d0e2f571a5]
10/21/2019, 10:24:02, INFO, Replacing instances to Utilize vacant RIs. Instances to replace: [i-0d65289d0e2f571a5]
10/21/2019, 10:45:52, INFO, Reserved instances launched: Utilizing Reserved Instances: i-0f95b3acd9cc3996e
10/21/2019, 10:45:52, INFO, Instances: [i-0f95b3acd9cc3996e] have been launched
10/21/2019, 10:47:37, INFO, Spotinst Kubernetes Controller, Instances Launched. Launched 1 instances. Ids: [sir-mcb89dqp]  (view details)
10/21/2019, 10:47:37, INFO, Instances: [i-06924a11b7039aa54] have been launched
10/21/2019, 10:47:37, INFO, Kubernetes Autoscaler, Scale Up Initiated due to pending for schedule Pods. Required Resources: 13000.0 of
millicpu. and 59392.0 of Mib RAM. Scale is restricted to Availability Zones: ["us-west-2a","us-west-2b","us-west-2c"]
10/21/2019, 10:50:02, INFO, Instances: [i-02ead6293c47adebf] have been launched
10/21/2019, 10:50:02, INFO, Spotinst Kubernetes Controller, Instances Launched. Launched 1 instances. Ids: [sir-kf78arvm]  (view details)
10/21/2019, 10:50:02, INFO, Kubernetes Autoscaler, Scale Up Initiated due to pending for schedule Pods. Required Resources: 11500.0 of
millicpu. and 34816.0 of Mib RAM. Scale is restricted to Availability Zones: ["us-west-2a","us-west-2b","us-west-2c"]
10/21/2019, 10:53:47, INFO, Spotinst Kubernetes Controller, Instances Launched. Launched 1 instances. Ids: [sir-kzviab9n]  (view details)
10/21/2019, 10:53:47, INFO, Instances: [i-0503103fae6c2350b] have been launched
10/21/2019, 10:53:47, INFO, Kubernetes Autoscaler, Scale Up Initiated due to pending for schedule Pods. Required Resources: 6750.0 of
millicpu. and 18432.0 of Mib RAM. Scale is restricted to Availability Zones: ["us-west-2a","us-west-2b","us-west-2c"]
10/21/2019, 11:01:14, INFO, Kubernetes Autoscaler, Scale down activity - Detached 2 instances Instances: [i-02ead6293c47adebf, i-
0f95b3acd9cc3996e]  (view details)
10/21/2019, 11:01:14, INFO, Instances: [i-02ead6293c47adebf,i-0f95b3acd9cc3996e] have been detached. Reason: Auto scale
10/21/2019, 11:07:13, INFO, Instances: [i-06924a11b7039aa54] have been detached. Reason: Auto scale
10/21/2019, 11:07:13, INFO, Kubernetes Autoscaler, Scale down activity - Detached 1 instances Instances: [i-06924a11b7039aa54]  (view
details)
10/21/2019, 11:24:02, INFO, Replacing instances to Utilize vacant RIs. Instances to replace: [i-0e89741acbf3b7d20]
10/21/2019, 12:24:02, INFO, Replacing instances to Utilize vacant RIs. Instances to replace: [i-0e89741acbf3b7d20]
10/21/2019, 13:24:02, INFO, Replacing instances to Utilize vacant RIs. Instances to replace: [i-0e89741acbf3b7d20]
10/21/2019, 14:24:02, INFO, Replacing instances to Utilize vacant RIs. Instances to replace: [i-0e89741acbf3b7d20]
```

Scale Behavior – Wavefront UI



-1H 2019-10-21 10:29:48 AM to 2019-10-21 11:30:25 AM

Compare OFF

Timezone [UTC -07:00] AMERICA/LOS_ANGELES (PDT)

Cluster eks-wavefront-prod-2

Mon Oct 21 2019 10:40:59AM -0700

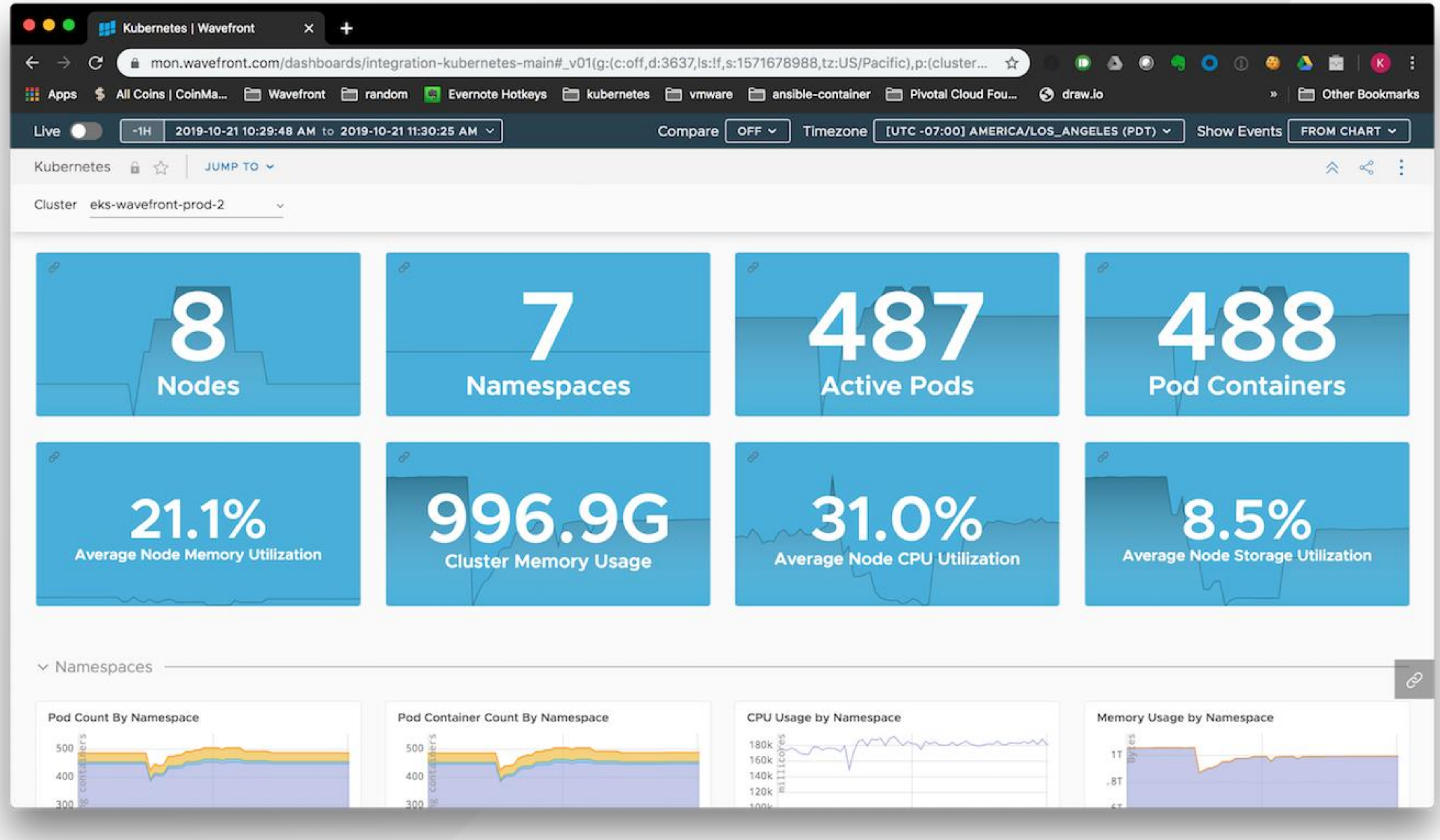
Source

	Source	
🟧	ip-10-211-180-43.wavefront-prod-2.k8s.wavefront.com	1.000
🟥	ip-10-211-148-87.wavefront-prod-2.k8s.wavefront.com	1.000
🟪	ip-10-211-111-206.wavefront-prod-2.k8s.wavefront.com	0.000
🟨	ip-10-211-168-67.wavefront-prod-2.k8s.wavefront.com	1.000
🟦	ip-10-211-153-147.wavefront-prod-2.k8s.wavefront.com	1.000
🟩	ip-10-211-141-19.wavefront-prod-2.k8s.wavefront.com	1.000
🟨	ip-10-211-117-243.wavefront-prod-2.k8s.wavefront.com	1.000
🟧	ip-10-211-137-185.wavefront-prod-2.k8s.wavefront.com	0.000
🟪	ip-10-211-96-197.wavefront-prod-2.k8s.wavefront.com	1.000
🟧	ip-10-211-25-216.wavefront-prod-2.k8s.wavefront.com	0.000
🟪	ip-10-211-177-83.wavefront-prod-2.k8s.wavefront.com	0.000
🟪	ip-10-211-119-207.wavefront-prod-2.k8s.wavefront.com	1.000

Source

```
count(ts("kubernetes.node.memory.usage", cluster="${cluster}"), sources)
```

Bring Your Metrics Front and Center



Kubernetes | Wavefront

mon.wavefront.com/dashboards/integration-kubernetes-main#_v01(g:(c:off,d:3637,ls:!f,s:1571678988,tz:US/Pacific),p:(cluster...

Live -1H 2019-10-21 10:29:48 AM to 2019-10-21 11:30:25 AM

Compare OFF

Timezone [UTC -07:00] AMERICA/LOS_ANGELES (PDT)

Show Events FROM CHART

Cluster eks-wavefront-prod-2

8 Nodes

7 Namespaces

487 Active Pods

488 Pod Containers

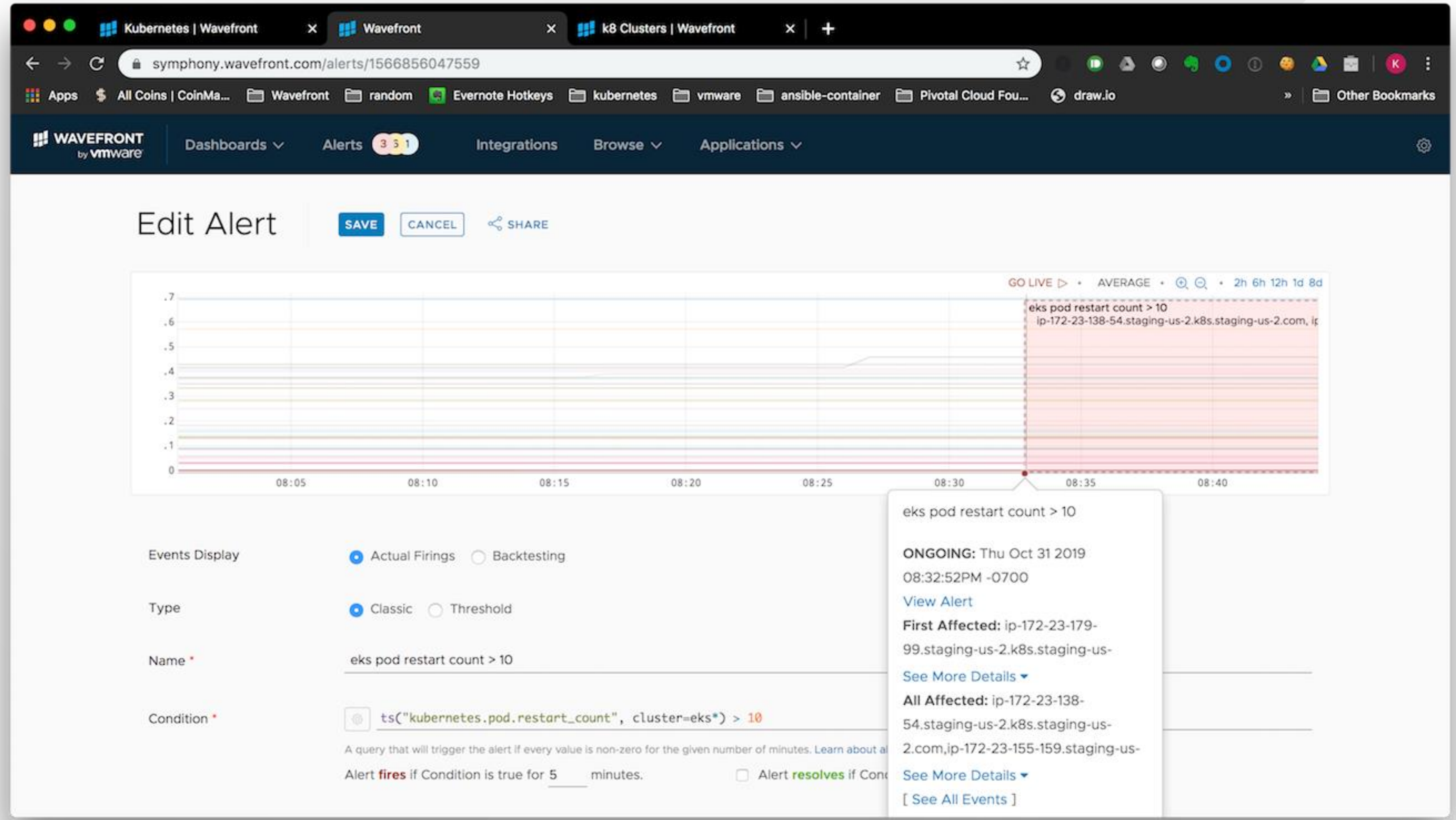
21.1% Average Node Memory Utilization

996.9G Cluster Memory Usage

31.0% Average Node CPU Utilization

8.5% Average Node Storage Utilization

Intelligent Alerting



Edit Alert

Events Display ● Actual Firings ○ Backtesting

Type ● Classic ○ Threshold

Name * eks pod restart count > 10

Condition *

```
ts("kubernetes.pod.restart_count", cluster=eks*) > 10
```

A query that will trigger the alert if every value is non-zero for the given number of minutes. Learn about al

Alert fires if Condition is true for 5 minutes.

□ Alert resolves if Conc

eks pod restart count > 10

ONGOING: Thu Oct 31 2019 08:32:52PM -0700

View Alert

First Affected: ip-172-23-179-99.staging-us-2.k8s.staging-us-

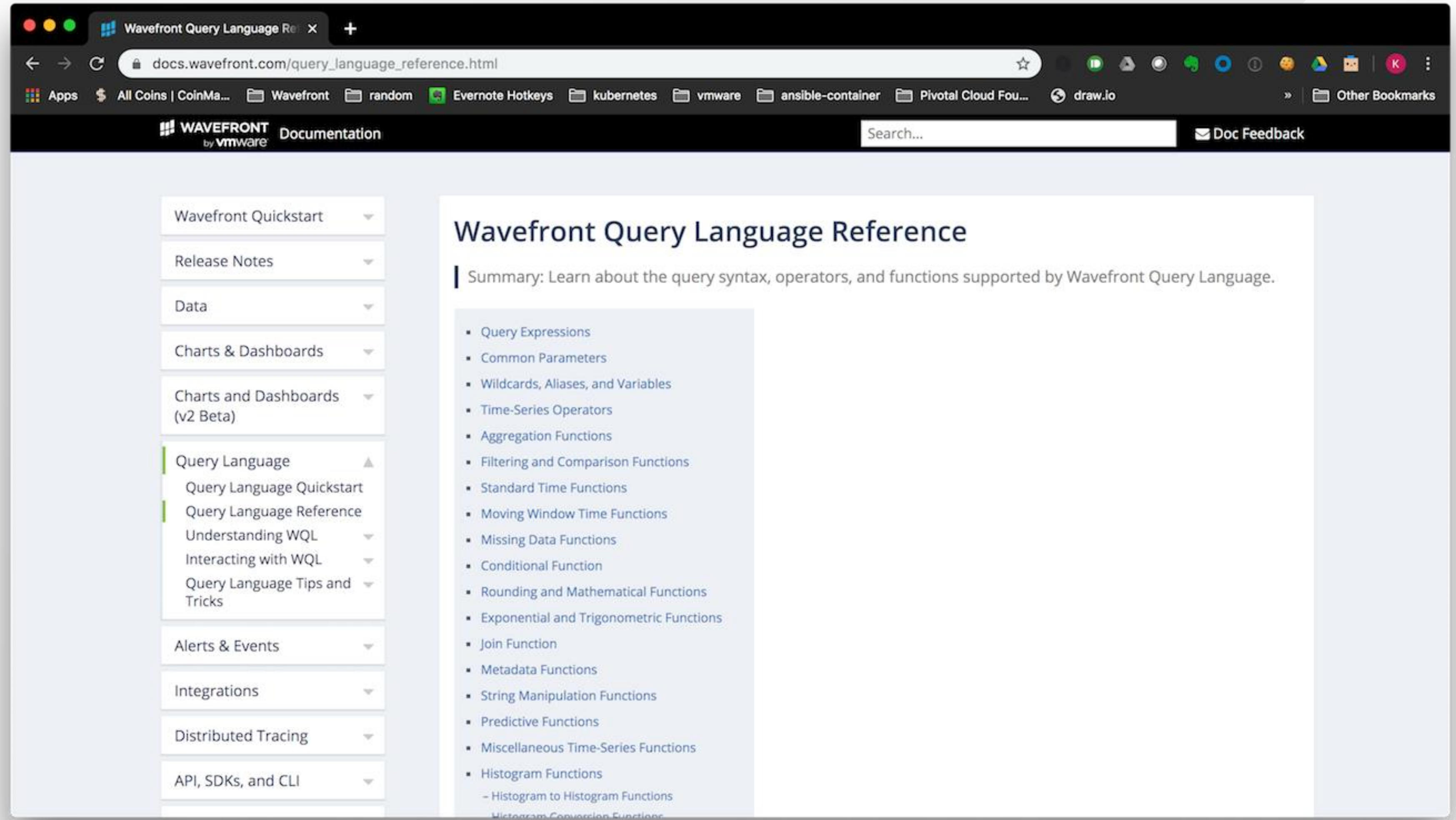
See More Details ▾

All Affected: ip-172-23-138-54.staging-us-2.k8s.staging-us-2.com,ip-172-23-155-159.staging-us-

See More Details ▾

[See All Events]

Robust Wavefront Query Language

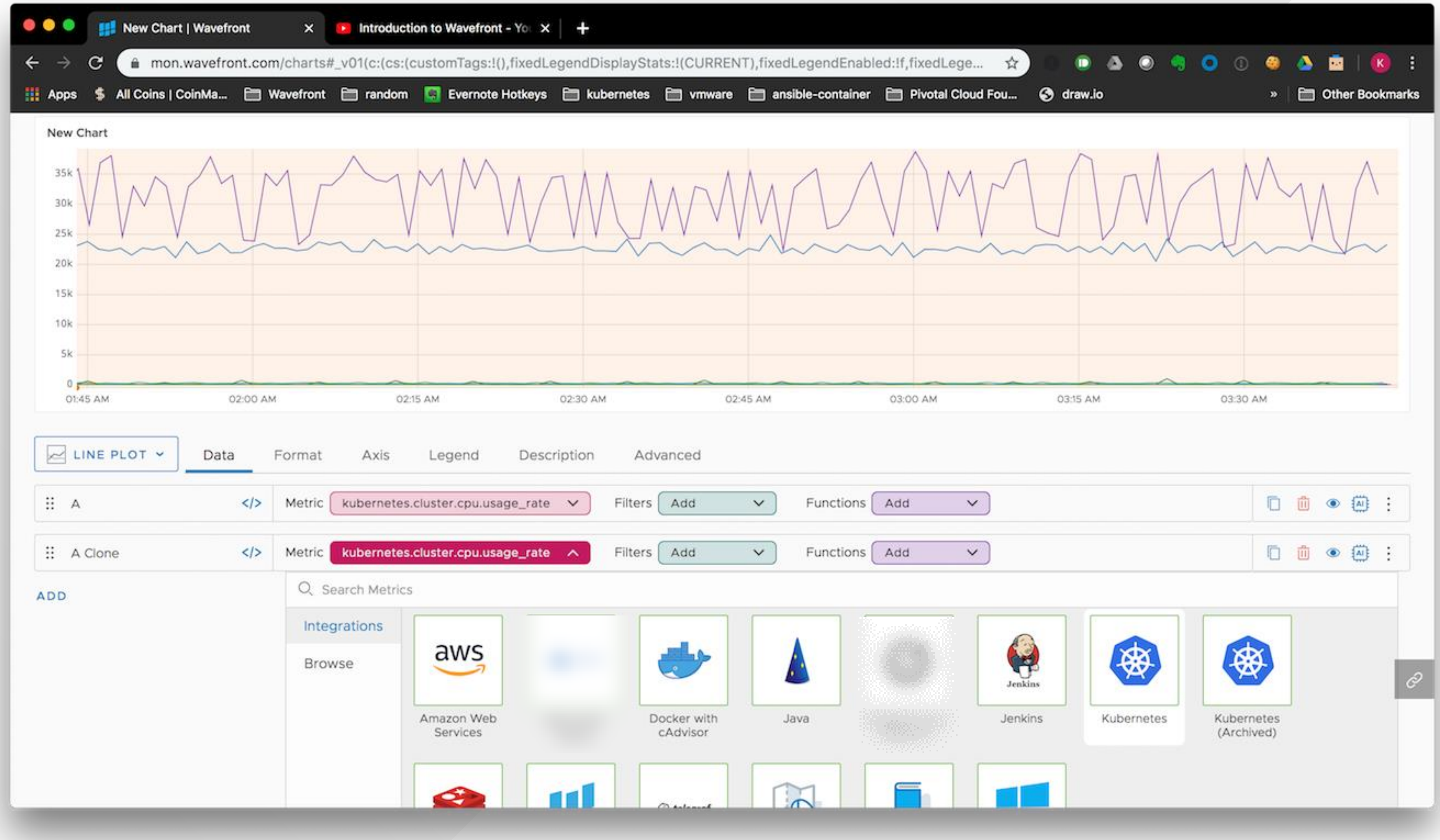


Wavefront Query Language Reference

Summary: Learn about the query syntax, operators, and functions supported by Wavefront Query Language.

- Query Expressions
- Common Parameters
- Wildcards, Aliases, and Variables
- Time-Series Operators
- Aggregation Functions
- Filtering and Comparison Functions
- Standard Time Functions
- Moving Window Time Functions
- Missing Data Functions
- Conditional Function
- Rounding and Mathematical Functions
- Exponential and Trigonometric Functions
- Join Function
- Metadata Functions
- String Manipulation Functions
- Predictive Functions
- Miscellaneous Time-Series Functions
- Histogram Functions
  - Histogram to Histogram Functions
  - Histogram Conversion Functions

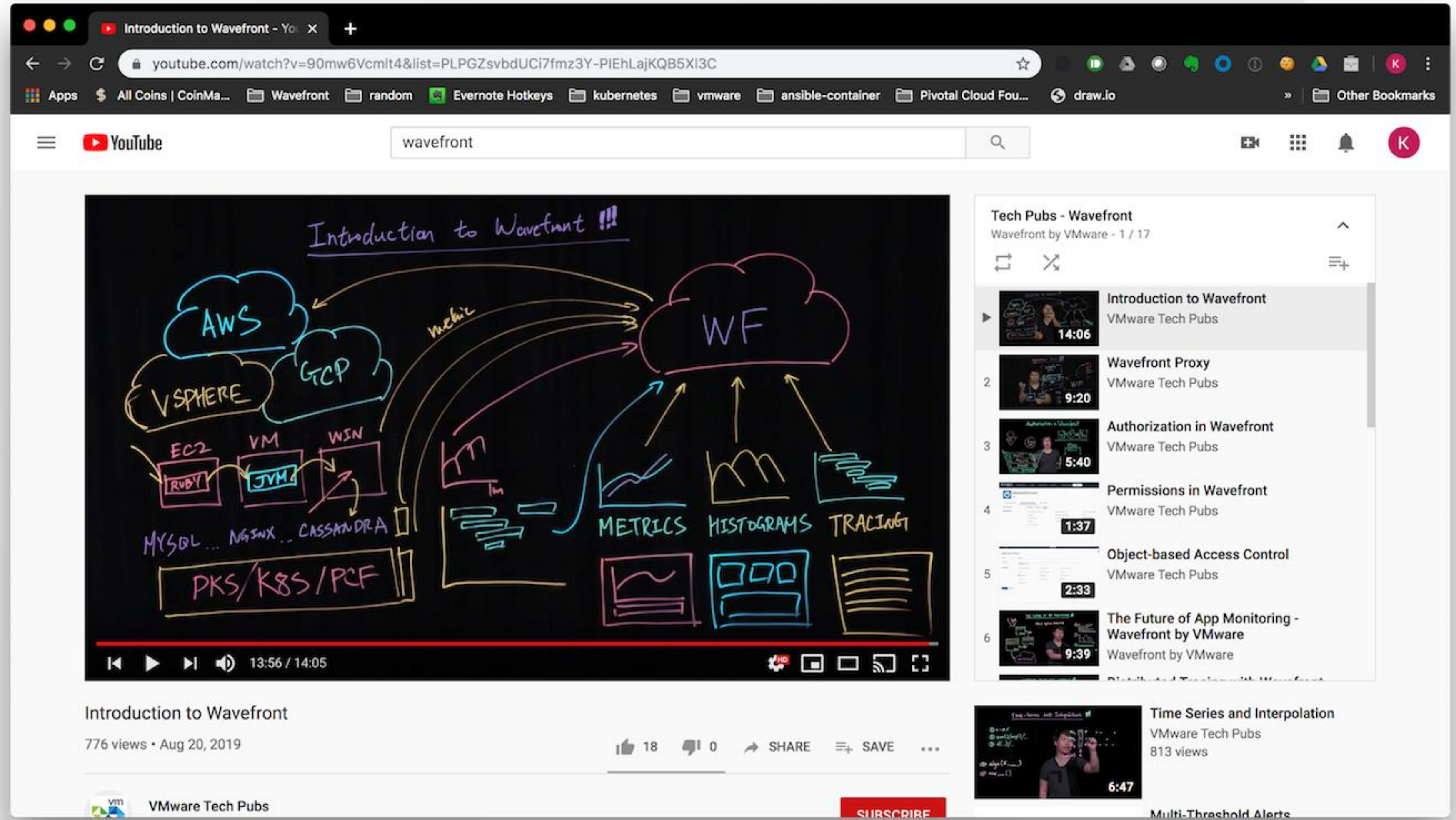
Interactive Query Builder



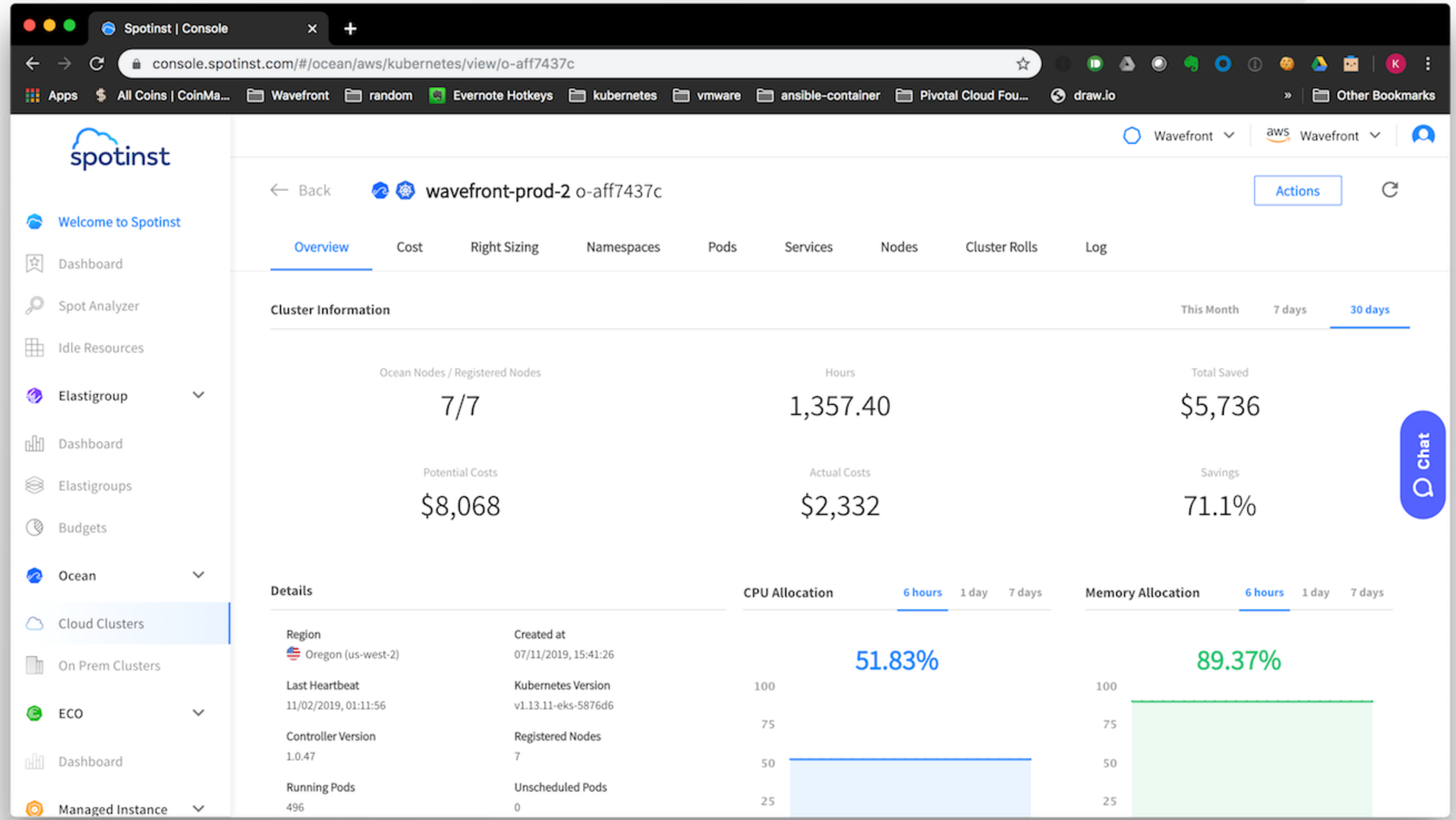
A </> Metric kubernetes.cluster.cpu.usage_rate Filters Add Functions Add

A Clone </> Metric kubernetes.cluster.cpu.usage_rate Filters Add Functions Add

Videos!


Introduction to Wavefront - Yo
youtube.com/watch?v=90mw6Vcmlt4&list=PLPGZsvbdUCi7fmz3Y-PIEhLajKQB5XI3C
Apps
All Coins | CoinMa...
Wavefront
random
Evernote Hotkeys
kubernetes
vmware
ansible-container
Pivotal Cloud Fou...
draw.io
Other Bookmarks
YouTube
wavefront
Introduction to Wavefront !!!
AWS
GCP
VSPHERE
EC2
VM
WIN
RUBY
JVM
MYSQL
NGINX
CASSANDRA
PKS/K8S/PCF
metric
WF
METRICS
HISTOGRAMS
TRACING
13:56 / 14:05
Tech Pubs - Wavefront
Wavefront by VMware - 1 / 17
Introduction to Wavefront
VMware Tech Pubs
14:06
2
Wavefront Proxy
VMware Tech Pubs
9:20
3
Authorization in Wavefront
VMware Tech Pubs
5:40
4
Permissions in Wavefront
VMware Tech Pubs
1:37
5
Object-based Access Control
VMware Tech Pubs
2:33
6
The Future of App Monitoring - Wavefront by VMware
Wavefront by VMware
9:39
Introduction to Wavefront
776 views • Aug 20, 2019
18
0
SHARE
SAVE
VMware Tech Pubs
SUBSCRIBE
Time Series and Interpolation
VMware Tech Pubs
813 views
6:47
Multi-Threshold Alerts

30-Day Savings on a Single Cluster



wavefront-prod-2 o-aff7437c

Cluster Information

Ocean Nodes / Registered Nodes
7/7

Hours
1,357.40

Total Saved
$5,736

Potential Costs
$8,068

Actual Costs
$2,332

Savings
71.1%

Details

Region
Oregon (us-west-2)

Created at
07/11/2019, 15:41:26

Last Heartbeat
11/02/2019, 01:11:56

Kubernetes Version
v1.13.11-eks-5876d6

Controller Version
1.0.47

Registered Nodes
7

Running Pods
496

Unscheduled Pods
0

CPU Allocation
51.83%

Memory Allocation
89.37%

Wavefront & Spotinst

1

Unified Full Stack View

30+%

Reduction in Tooling Complexity

10x

Earlier Issue Detection

5x

Lower Prices than Traditional APM

100B+

Data Points Ingested Per Day (at Scale)

Wavefront & Spotinst

1

Unified Full Stack View

30+%

Reduction in Tooling Complexity

10x

Earlier Issue Detection

5x

Lower Prices than Traditional APM

100B+

Data Points Ingested Per Day (at Scale)

Wavefront & Spotinst

1

Unified Full Stack View

30+%

Reduction in Tooling Complexity

10x

Earlier Issue Detection

5x

Lower Prices than Traditional APM

100B+

Data Points Ingested Per Day (at Scale)

Wavefront & Spotinst

1

Unified Full Stack View

30+%

Reduction in Tooling Complexity

10x

Earlier Issue Detection

5x

Lower Prices than Traditional APM

100B+

Data Points Ingested Per Day (at Scale)

Wavefront & Spotinst

1

Unified Full Stack View

30+%

Reduction in Tooling Complexity

10x

Earlier Issue Detection

5x

Lower Prices than Traditional APM

100B+

Data Points Ingested Per Day (at Scale)

Wavefront & Spotinst

1

Unified Full Stack View

30+%

Reduction in Tooling Complexity

10x

Earlier Issue Detection

5x

Lower Prices than Traditional APM

100B+

Data Points Ingested Per Day (at Scale)

Thank you!

Kevin McGrath

kevin@spotinst.com
www.linkedin.com/in/mcgrathk
Twitter :@catlgrep

Kai Paro

kparo@vmware.com
www.linkedin.com/in/kaiparo
Twitter: @K__Paro

The image part with relationship ID rId3 was not found in the file.

Please complete the session survey in the mobile app.

The image part with relationship ID rId3 was not found in the file.