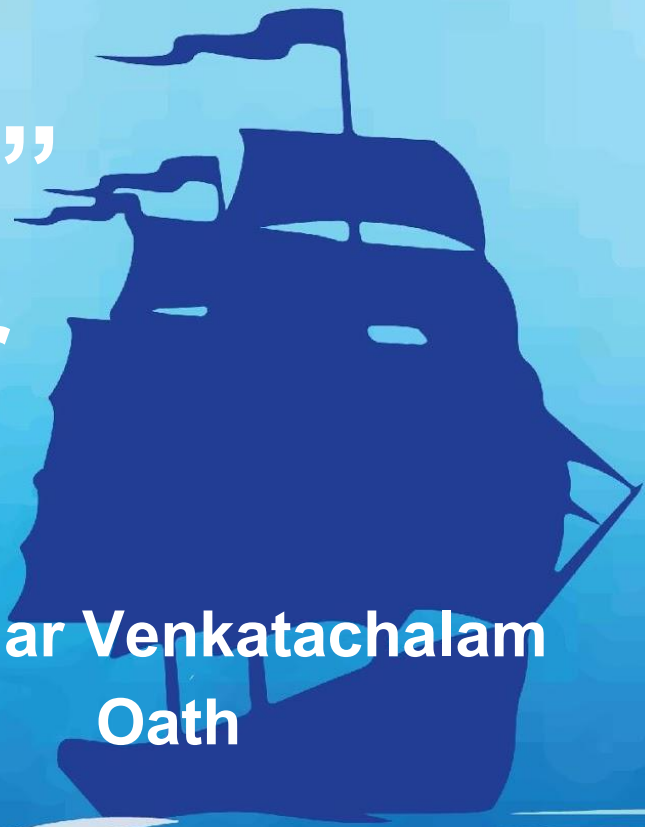
KubeCon | CloudNativeCon

Europe 2018

“Break and Recover” Kubernetes Cluster

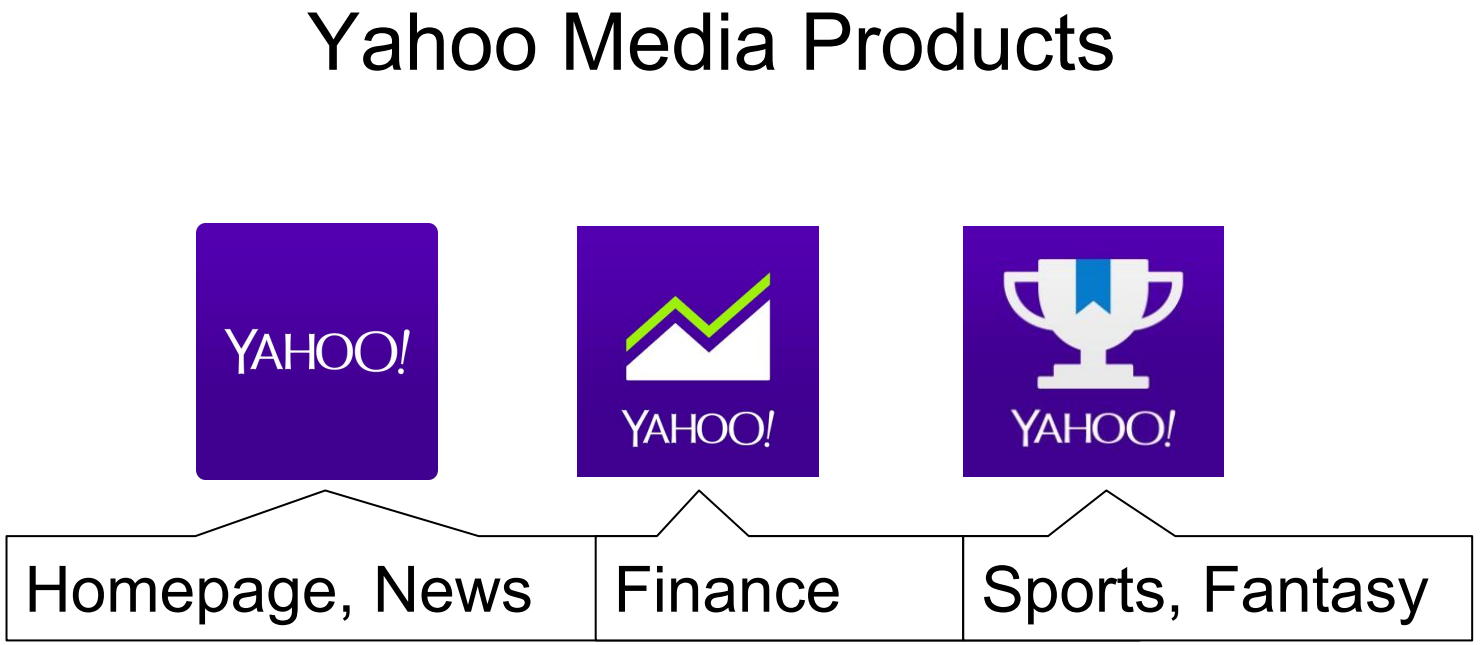


Suresh Visvanathan
Oath

Nandhakumar Venkatachalam
Oath

Team

- Core Platform
  - Team powering all Yahoo Media Products

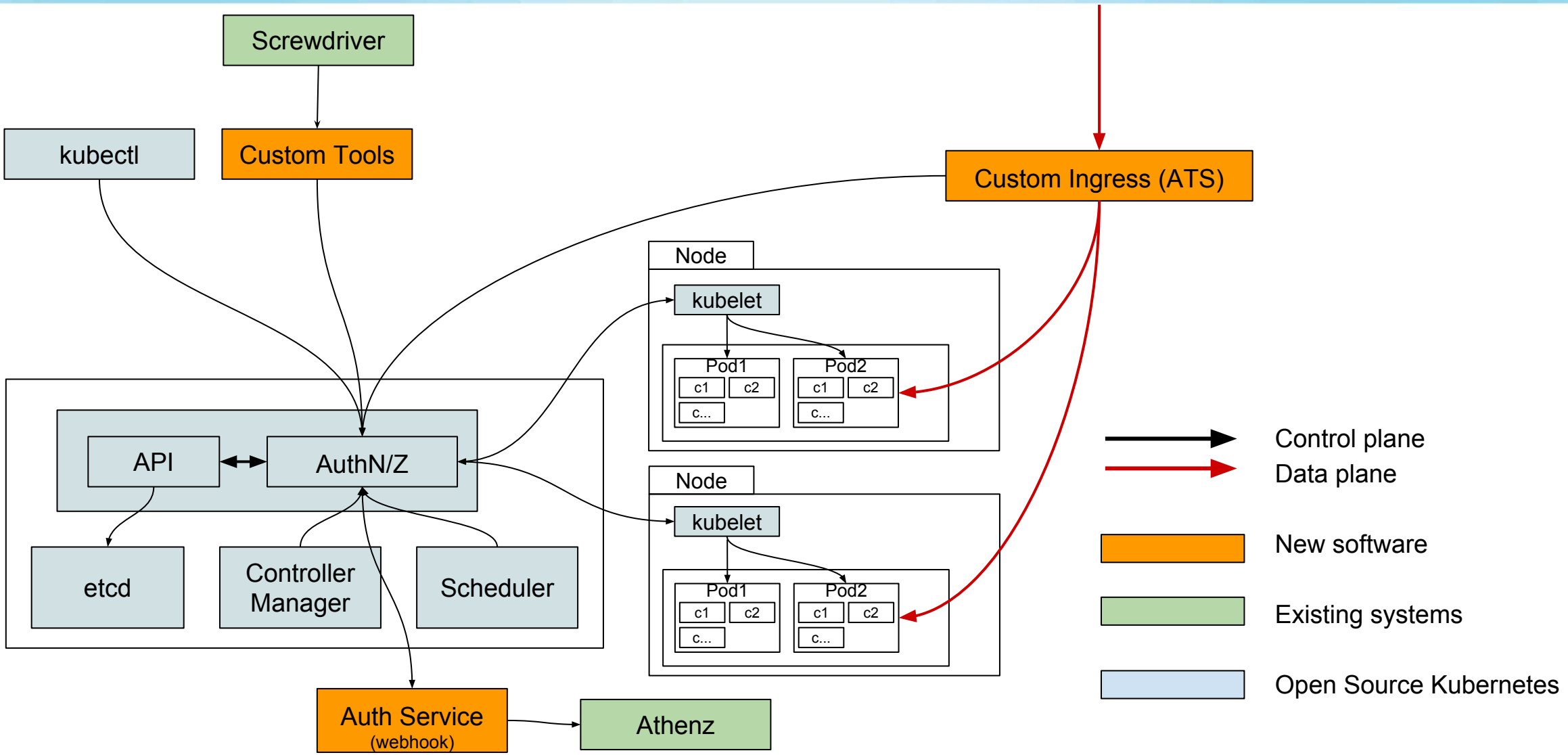


Team

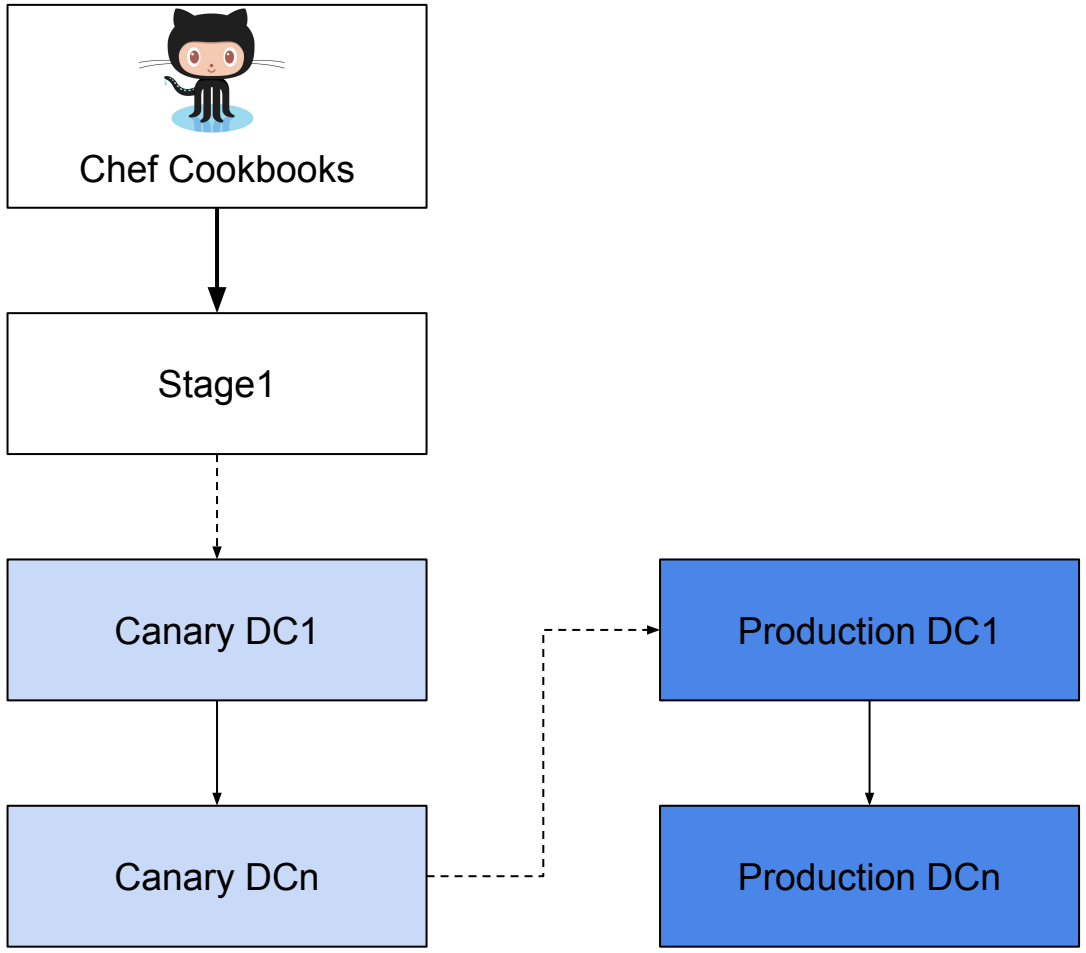
Kubernetes at Oath

- 12 independent clusters across 6 data centers globally
- 2K+ Worker Nodes
- 12K+ Pods
- 50K+ Containers
- 200+ Application deployments
  - Mostly stateless workloads
  - Few stateful workloads
- Peak requests/sec - 400K+

Kubernetes at Oath



Cluster Pipeline



- Chef cookbook with cluster pkg and service configs
- Changes are baked in stages
- Ability to override package settings specific to an env

On-Prem K8s Experience

- Kubernetes is awesome
- Many interdependent components
  - Etcd, Controller, Scheduler, Api Server, Nodes (kubelet), Docker and many control loops

Oath <3s K8s

Namespace

- Multitenant with namespace isolation
- 100+ namespaces
- `kubectl delete -f /dir`
  - One of file in the directory was ns file.

Result :

- Cascading impact
- Everything within the namespace wiped out

Namespace Protection

- Retrigger CI/CD pipeline for Quick Recovery.
- Admission Controller
  - k8s-namespace-guard

```
apiVersion: admissionregistration.k8s.io/v1beta1
kind: ValidatingWebhookConfiguration
metadata:
  name: admission-hook-config
webhooks:
- name: ns-guard.kube-bag.hookcfg
  rules:
  - apiGroups:
    - ""
    apiVersions:
    - "v1"
    operations:
    - DELETE
    resources:
    - "namespaces"
  failurePolicy: Fail
  clientConfig:
    caBundle: "@caBundle"
    service:
      name: ns-guard
      namespace: kube-bag
```

Ingress - Usurping of domain name

- Single Ingress Controller per cluster
- Ingress resource to define

  hostname/path/backend

Domain name collision


```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  labels:
    app: generic-app1-production-us-west
    environment: production
    appName: generic-app1
  name: generic-app1-production-us-west
  namespace: generic-app1-k8s
  annotations:
    ports: "80,443"
    aliases: "kubecon.media.yahoo.com"
    default_domain: "generic-app1.production.us-west.yahoo.com"
spec:
  backend:
    serviceName: generic-app1-production-us-west
    servicePort: 8080
```

```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  labels:
    app: generic-app2-production-us-west
    environment: production
    appName: generic-app2
  name: generic-app2-production-us-west
  namespace: generic-app2-k8s
  annotations:
    ports: "80,443"
    aliases: "kubecon.media.yahoo.com"
    default_domain: "generic-app2.production.us-west.yahoo.com"
spec:
  backend:
    serviceName: generic-app2-production-us-west
    servicePort: 8080
```

Ingress - domain name protection

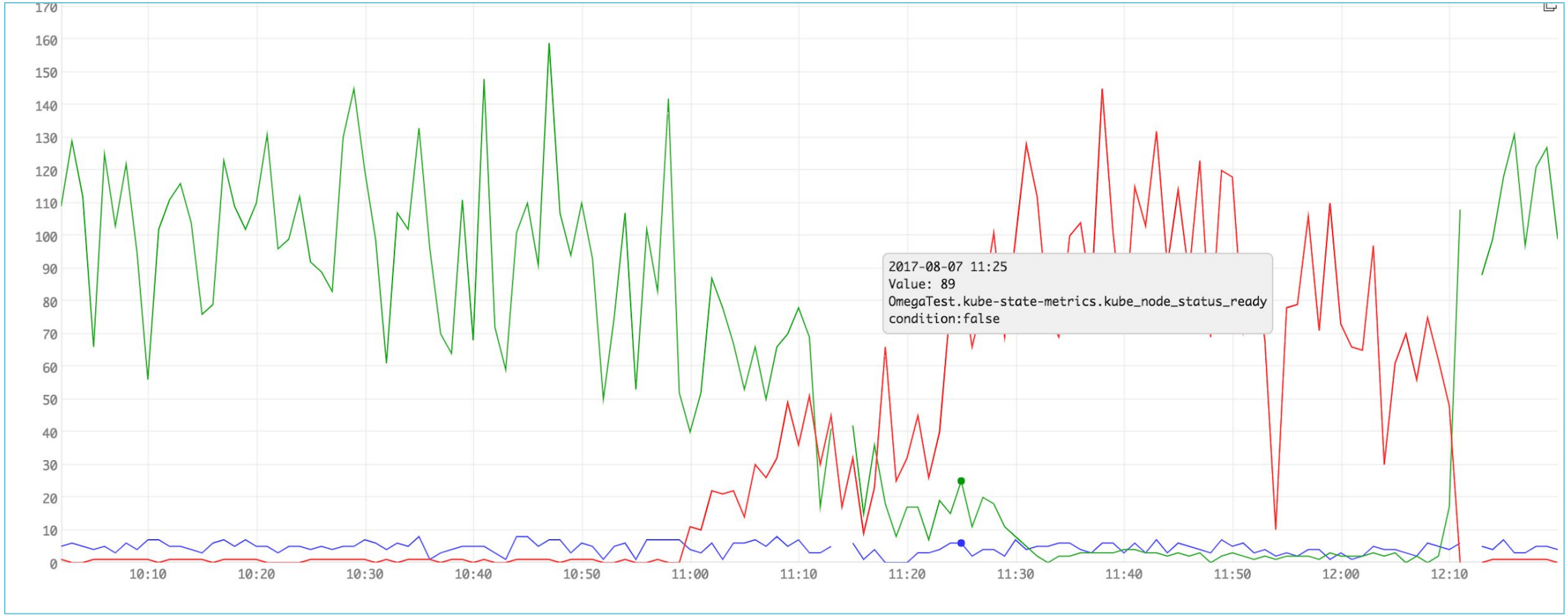
- Most Ingress does RR between 2 backend
- Admission Controller
  - k8s-ingress-claim

```
apiVersion: admissionregistration.k8s.io/v1beta1
kind: ValidatingWebhookConfiguration
metadata:
  name: admission-hook-config
webhooks:
- name: ingress-claim.kube-bag.hookcfg
  rules:
  - apiGroups:
    - "extensions"
    apiVersions:
    - "v1beta1"
    operations:
    - CREATE
    - UPDATE
    resources:
    - "ingresses"
  failurePolicy: Fail
  clientConfig:
    caBundle: "@caBundle"
    service:
      name: ingress-claim
      namespace: kube-bag
```

K8s Nodes

- Deploy triggered for K8s nodes.
- Transient Cgroup file cleanup failure
  - https://github.com/kubernetes/kubernetes/issues/43856
- After deployed, we had 100% Node are in “NotReady” state

K8s Nodes


170
160
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
10:10
10:20
10:30
10:40
10:50
11:00
11:10
11:20
11:30
11:40
11:50
12:00
12:10
2017-08-07 11:25
Value: 89
OmegaTest.kube-state-metrics.kube_node_status_ready
condition:false

K8s Nodes

- Nodes marked ‘NotReady’
  - Added to eviction Queue
- Pod eviction process kick in
- 50% nodes lost
  - Cluster enters into ‘**Partial Disruption mode**’
  - Throttles the rate of eviction
- 100% nodes lost
  - Clusters enters into ‘**Full Disruption mode**’
  - Stops Eviction

K8s Nodes

- Quick Recover is to fail out of the data center
- Rolling update with 30% concurrency
  - Post functional test to certify the node.

ETCD

- Etcd upgrade
- K8s API was up and running
- Etcd server started pointing to empty directory

Result

- Kubelet synchronizes with Etcd state and self evicted all pods across fleet

ETCD Recovery

- Recovered by pointing to right directory
- Database snapshot backup and restore

TLS Certificate Refresh

- Periodic Refresh of TLS Certs for Control plane

- Restart all the components if root ca is changed
- If `--service-account-private-key-file` changed
  - All Service Accounts communication broken
  - Most control plane components came crashing
  - Identity provider failed
    - Causing all apps to fails

TLS Certificate Refresh

- Apiserver `--service-account-key-file` supports multi keys.
- Still SA need to be recreated or refreshed to get rid of old keys.

Kube DNS

- Application use kube-dns as a resolver.
  - If the dns policy is `ClusterFirst`

- Kube-DNS Pod overload can slow app performance
  - App with no cache can load kube-dns
- K8s node of kube-dns pod down
- Added latency for external dns name
  - .5 traversal for external domain lookup

Kube DNS

- We run dnsmasq as a daemonset and set kubelet's to
  - `--cluster-dns to kubeletmachineIP`
  - `Tried tuning --pod-eviction-timeout (contr. flag)`
- Use of cluster proportional autoscaler

Keeping Up to Date

- Upgrade K8s and Dependencies
- Keep up with OS and kernel
  - NFS mount fails frequently, rpcbind upgrade
  - dbus daemon had fd leak, dockerd to fail
    - `dbus-daemon[1466]: Failed to close file descriptor: Could not close fd 1591`

Beware of OOM

- OOM Killer could cause a machine hang.
- 9600 baud rate limits on the console.
- Log less verbose oom

```
kernel.printk = 5 5
vm.oom_dump_tasks = 0
vm.overcommit_memory #consider 0
#kernel doesn’t aware of process container cgroup limit
```

Out Of Resource Handling

- Configuring out of resource handling to handle node pressure

```
--eviction-soft='memory.available<5%,nodefs.available<5%,
imagefs.available<5%,nodefs.inodesFree<5%'
```

- Enable log rotation in docker

```
daemon.json {
                "live-restore": true,
                "log-driver": "json-file",
                "log-opts": {  "max-size": "10g" }
               }
```

Gameday Testing

Control Plane	Node
Bring down an apiserver	BGP daemon (bird) goes down
Bring down the scheduler	Bring down Docker Daemon
Bring down the controller manager	Bring down kubelet
Bring down one etcd node	Bring down kube-proxy
Break etcd quorum	Node reboot
ETCD disaster recovery	Switch down
	20% - 30% nodes down
	Simulate disk pressure, storage full

Links

Authorization plugins

- https://github.com/yahoo/k8s-athenz-identity
- https://github.com/yahoo/k8s-athenz-webhook
- https://github.com/yahoo/athenz

Admission control plugins

- https://github.com/yahoo/k8s-namespace-guard
- https://github.com/yahoo/k8s-ingress-claim

Geeking out K8s @ Oath

- K8s Bay Area Meetup - https://goo.gl/Z1SuDQ

sureshv@oath.com
https://www.linkedin.com/in/sureshvisvanathan
@sureshvisvanath

Thank you

KubeCon

CloudNativeCon

Europe 2018

Q & A