KubeCon | CloudNativeCon

North America 2017

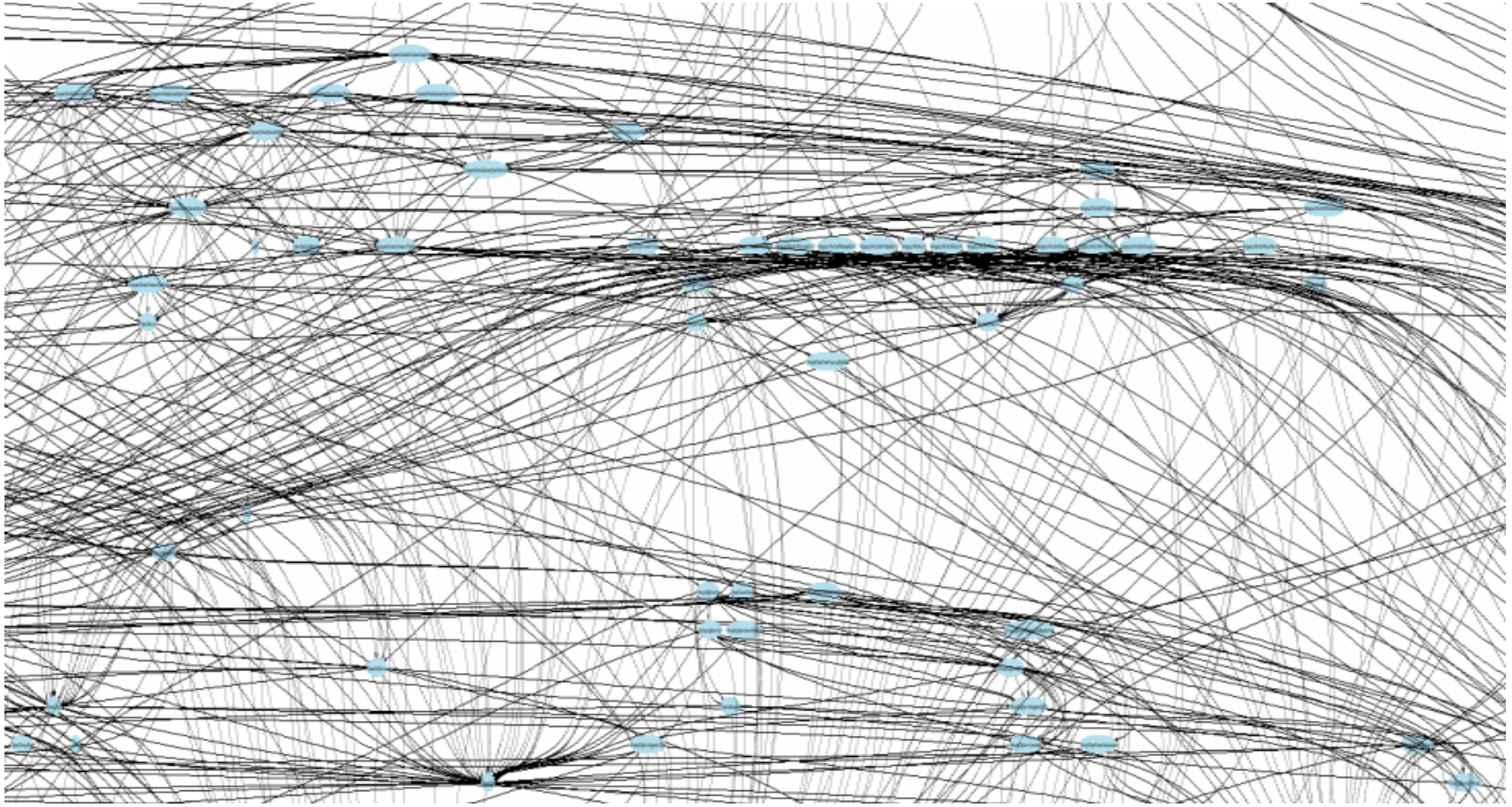
Pushing the Limits with GAME OF THRONES

Illya Chekrygin, Senior Staff Engineer, HBO
Zihao Yu, Senior Staff Engineer, HBO

HBO

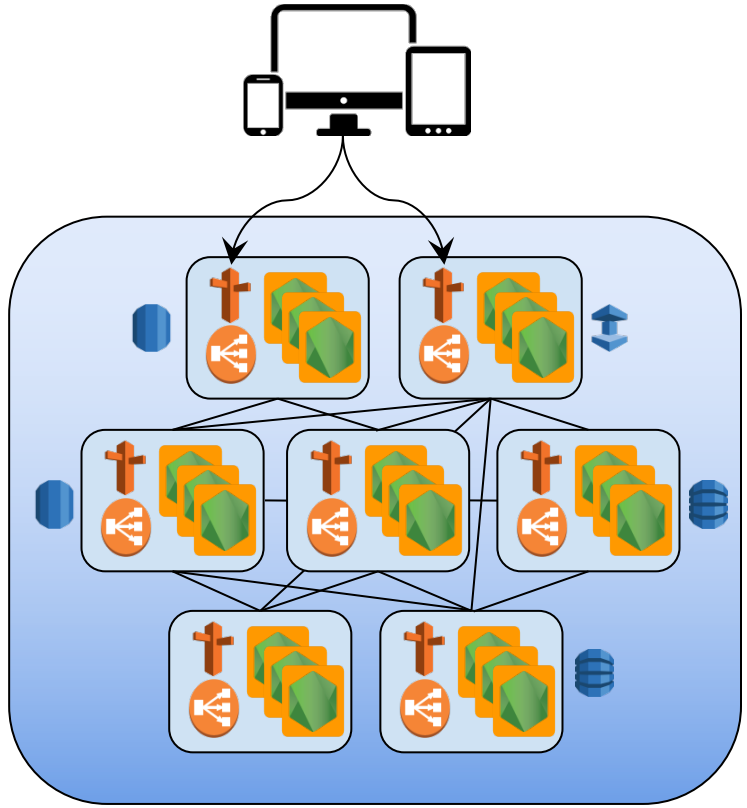
- About HBO
- About HBO Digital Products
- Zihao Yu
  - zihao.yu@hbo.com
  - @zihaoyu  Kubernetes kubernetes.slack.com
- Illya Chekrygin
  - illya.chekrygin@hbo.com
  - @ichekrygin  Kubernetes kubernetes.slack.com

HBO Streaming Services

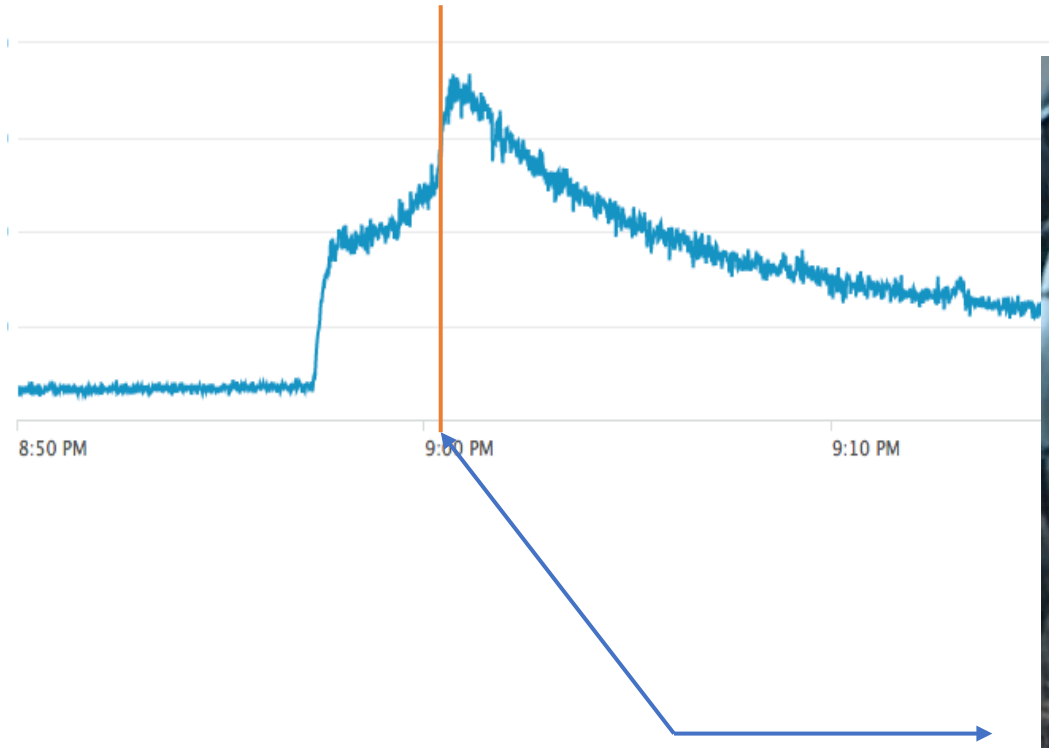


Streaming Services on EC2

- API Services in Nodejs
- Single Service = Single EC2 Instance
- ASG + Configuration
- Fronted by ELB
  - Home grown Service Discovery
- Route 53 for DNS



HBO Traffic Pattern



Problems

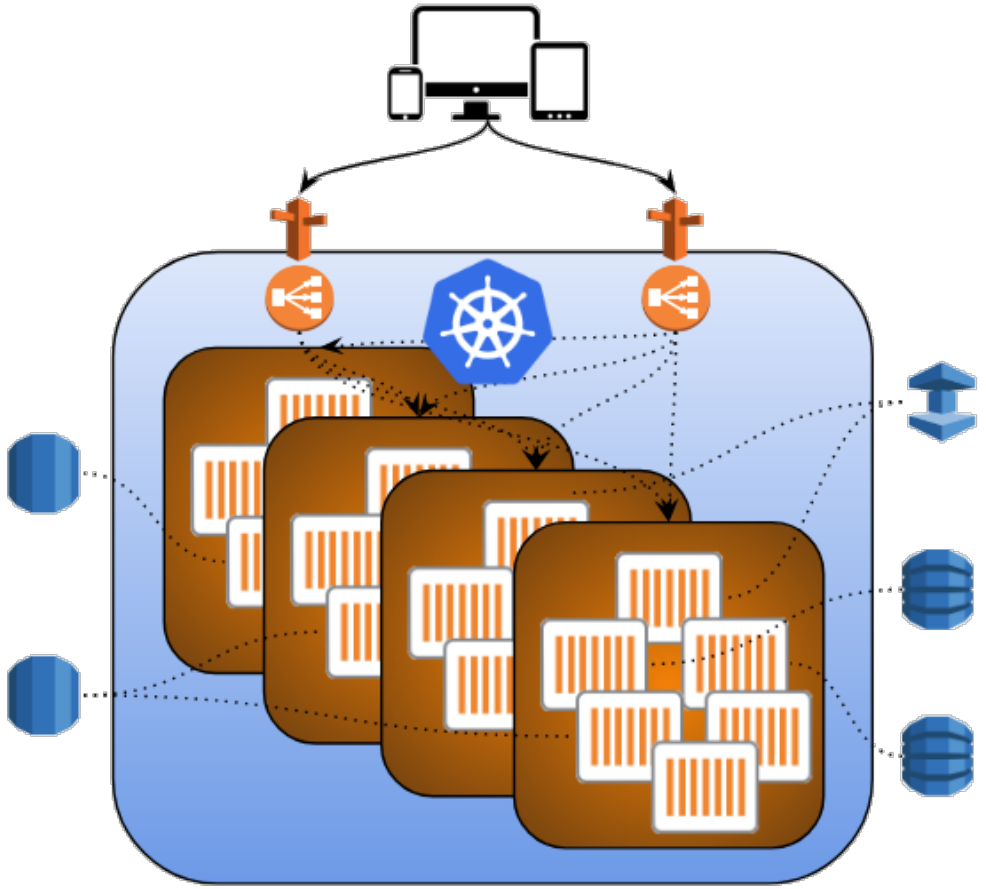
- Wasteful
  - Nodejs + EC2 = 50+% of CPU unutilized
  - Multiply by Overscaling
- On Demand Scaling Up/Down
  - Slow...
- Service to Service
  - You get an ELB!
  - You get an ELB!
  - Everybody gets ELB!

Limits

- EC2
  - IP addresses within Subnet
  - Instance Types
  - ELB’s
  - Autoscaling Groups and Configurations
- Auxiliary
  - Telemetry licensing (per instance)

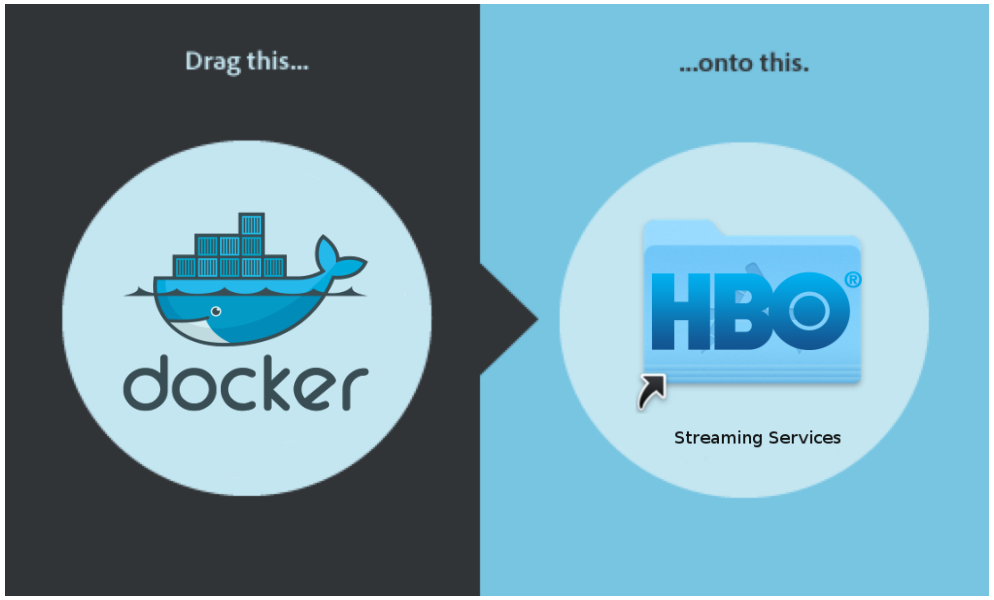
Why Kubernetes?

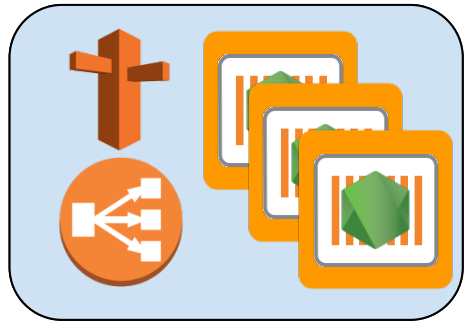
- Utilization
- Introspection
- Faster
  - Deployments and Rollbacks
  - Pod Autoscaling
- Safer
  - Rolling Updates
  - Configuration as code
- Batteries are included
  - DNS, SD, kubectl



Step 1

- End of 2015
  - 0 containerized services
- July 2017
  - GoT Season 7 Premiere
- Dockerize
  - Modify EC2 Deployments
  - Modify CI/CD Pipelines





Step Next

- Run Kubernetes
- But...

Kubernetes on AWS

- kube-up.sh and all the defaults
- Infrastructure Cluster
  - Jenkins CI/CD
- Service Cluster
- Basic setup is not enough...
  - Network
    - VPC + Route Tables (50)
    - Multi-AZ support

Kubernetes -Terraforming

- Custom Terraform templates and cloud-init
  - Before kube-aws, kops, or kubespray projects
  - Use existing infrastructure as Terraform variables
    - VPC Subnets
    - Security Groups

Kubernetes -Terraforming

- Features
  - HA multi-masters
  - Masters and minions are in Autoscaling Groups
  - Masters and minions are multi-AZ
  - OIDC authentication
    - CoreOS Dex
    - Github OAuth

Terraform Template For A Cluster

```
provider "aws" {
  region = "us-west-2"
}

module "kubernetes_master" {
  source =
    git@github.com:HBO/TF-Modules.git//k8s/master?ref=v1
}

module "kubernetes_minion" {
  source =
    git@github.com:HBO/TF-Modules.git//k8s/minion?ref=v1
}
```

Small Issues

- Prometheus pod rescheduled when cluster scaled down
- AWS insufficient capacity of our desired instance type

Terraform Template For A Cluster - v2

```
// Terraform module for minions
resource "aws_launch_configuration" "minion" {
  instance_type = "${var.instance_type}"
}
```

Terraform Template For A Cluster - v2

```
// Terraform module for minions
minion_taints = "${var.taints == "" ? "" :
  "--register-with-taints=${var.taints}"}"

// cloud-init
ExecStart=/opt/bin/kubelet ${minion_taints}
```

Terraform Template For A Cluster - v2

```
module "regular_minion" {
  source =
    git@github.com:HBO/TF-Modules.git//k8s/minion?ref=v2
  instance_type = "<main instance type>"
}
```

Terraform Template For A Cluster - v2

```
module "backup_minion" {
  source =
    git@github.com:HBO/TF-Modules.git//k8s/minion?ref=v2
  instance_type = "c4.8xlarge"
}
```

Terraform Template For A Cluster - v2

```
module "reserved_minion" {
  source =
    git@github.com:HBO/TF-Modules.git//k8s/minion?ref=v2
  taints = "reserved=true"
}
```

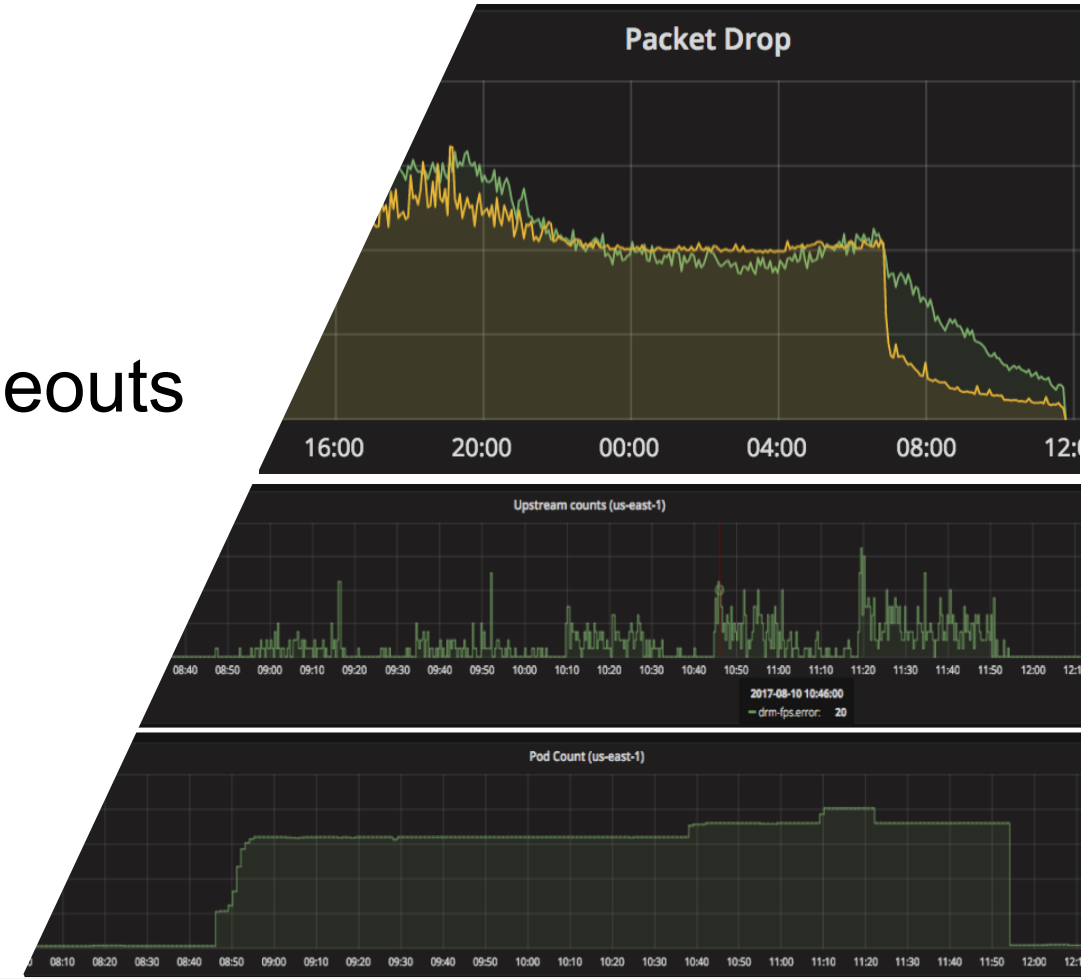
Terraform Template For A Cluster - v2

```
module "backup_minion" {
  source =
    git@github.com:HBO/TF-Modules.git//k8s/minion?ref=v2
  instance_type = "c4.8xlarge"
}

module "reserved_minion" {
  source =
    git@github.com:HBO/TF-Modules.git//k8s/minion?ref=v2
  taints = "reserved=true"
}
```

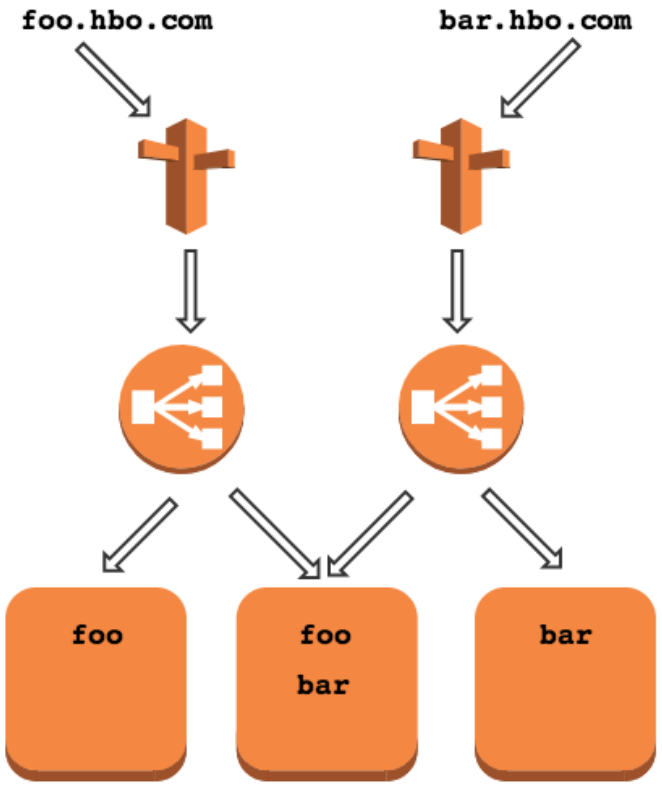
Networking - Flannel

- Simple Setup
- Scaling Up (Way Up)
  - Increased Latency and Timeouts
  - UDP Packet Drop
- Good for GoT Season 7
  - coreos/flannel#414
  - coreos/flannel#427



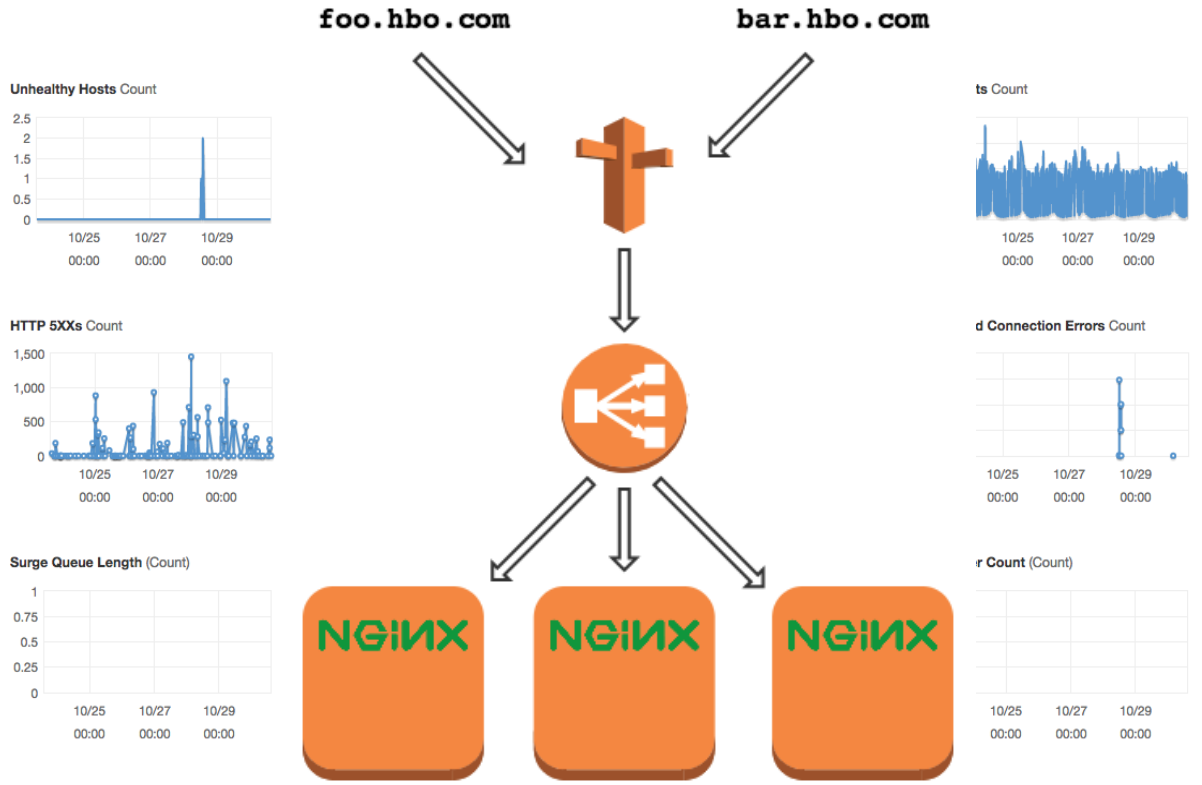
Networking - Service, Ingress and LBs

- `type:NodePort`
  - Limit 50 ELBs per ASG
  - Keep track of ELBs by yourself



Networking - Service, Ingress and LBs

- `type: ClusterIP` + Ingress
  - Whose 500s are those anyway?
  - Hard to keep up with burst traffic
  - Internal/External



Networking - Service, Ingress and LBs

- `type: LoadBalancer`
  - AWS API Rate Limits
  - Security Group Customization
    - kubernetes/kubernetes#49805

Networking - Service, Ingress and LBs

Environment	Choice
Production	NodePort + ELB + K8S SD
Non-Production	ClusterIP + Ingress + Shared ELB + K8S SD

Networking - kube-dns

- /etc/resolv.conf and ndots:5
  - ```
    search default.svc.cluster.local
    svc.cluster.local cluster.local us-west-
    2.compute.internal
    nameserver 10.5.0.10
    options ndots:5
    ```
    ```

Networking - kube-dns

- pgsql.backend.hbo.com =>
  - pgsql.backend.hbo.com.default.svc.cluster.local
  - pgsql.backend.hbo.com.svc.cluster.local
  - pgsql.backend.hbo.com.cluster.local
  - pgsql.backend.hbo.com.us-west-2.compute.internal
- kubernetes/kubernetes#33554

kube-dns Tuning

```
# max cache size (dnsmasq)

--cache-size=10000
```

kube-dns Tuning

```
# return NXDOMAIN for invalid queries (dnsmasq)
# format: --address=/<domain>/<ipaddr>

--address=
 /hbo.com.cluster.local/hbo.com.svc.cluster.loca
 l/hbo.com.production.svc.cluster.local/hbo.com.
 default.svc.cluster.local/hbo.com.kube-
 system.svc.cluster.local/hbo.com.us-west-
 2.compute.internal/
```

← NO <ipaddr> provided!

kube-dns Tuning

```
# internal nameserver (dnsmasq)

--server=/homeboxoffice.com/10.1.2.3#53
```

Telemetry

- Grafana
- Graphite + Statsd
- ELK + Fluentd
- Splunk Forwarder
- Prometheus + cAdvisor
  - Not so easy with 300 nodes @40 cores each + 20K containers
  - EBS
    - Rook

Were We Ready?

- Load Test
- Load Test More!
- Load Test the SHIT Out of It!!!
- Search, Report, Fix, Contribute

Our Advice

- Trust Yourself
  - Small increments towards the big goal
- Trust the Community
  - Strong
  - Active
  - Diverse
  - You won’t be stranded
- Watch HBO!

Kubernetes at HBO

Thank You!