Scaling Kubernetes to Thousands of Nodes Across Multiple Clusters (calmly)

BEN HUGHES • NOV 19 • KUBECON 2019

Agenda

- How we got here
- What the scaling limits are
- Scaling to multiple clusters
- How we manage our clusters
- Dealing with change

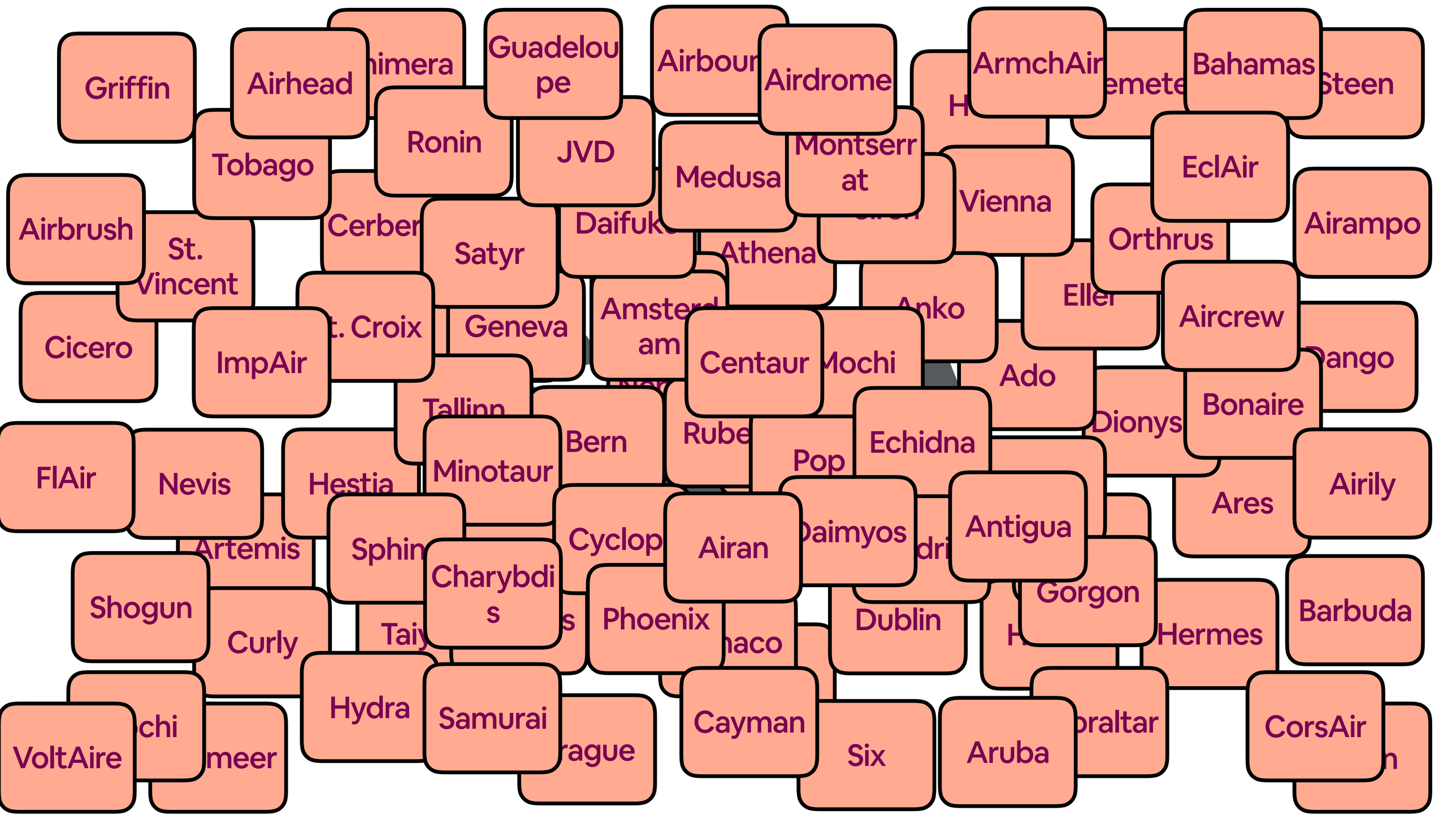
How we got here

Monolith

Monorail

Lover of Romance

SOA


Griffin
Airhead
himera
Guadelou
pe
Airbour
Airdrome
ArmchAir
Bahamas
emete
Steen
Tobago
Ronin
JVD
Medusa
Montserr
at
EclAir
Airbrush
St.
Vincent
Cerber
Satyr
Daifuk
Athena
Vienna
Orthrus
Airampo
Cicero
t. Croix
Geneva
Amsterd
am
Anko
Eller
Aircrew
ImpAir
Centaur
Mochi
Ado
Dionys
Bonaire
Tallinn
Bern
Rube
Pop
Echidna
FlAir
Nevis
Hestia
Minotaur
Ares
Airily
Artemis
Sphin
Cyclop
Airan
Daimyos
Antigua
Shogun
Charybdi
s
Phoenix
Dublin
Gorgon
Hermes
Barbuda
Curly
Taiy
naco
Hydra
Samurai
Cayman
Six
Aruba
oraltar
CorsAir
VoltAire
meer
ague

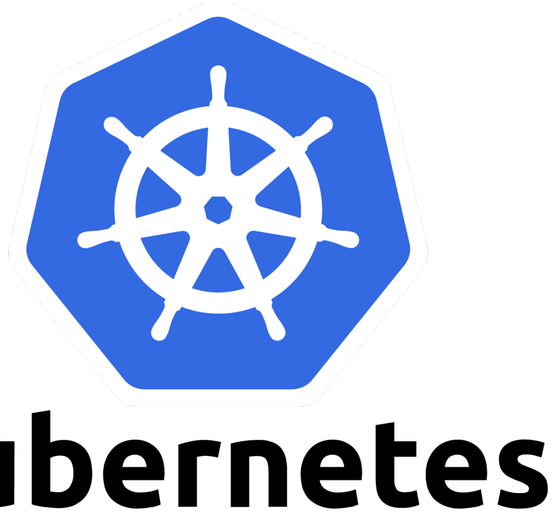
Many Touches

- Application code changes
- Chef changes
- Manage state in EC2
- Terraform for IAM Permissions?
  - (and the even worse things before we had that)
- Alerts, Monitoring
- Databases, probably
- So many pull requests

OneTouch

kubernetes

OneTouch



And of course a lot of other configuration and tooling to abstract and contain everything...

OneTouch

- One Pull Request
- One Review
- One Deploy
- to handle all app/config changes for the service

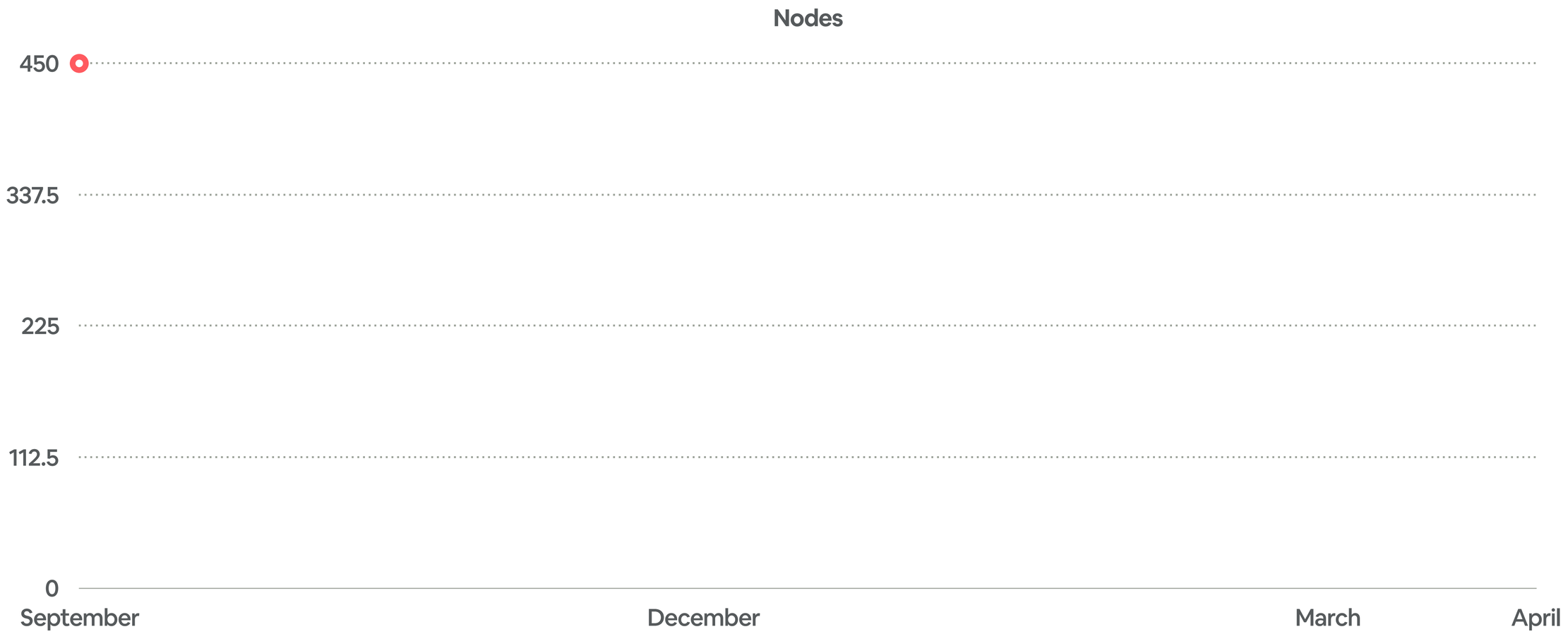
(Hundreds of services is probably still too many)

Migration

- Move all existing services from Chef/EC2 to OneTouch/ Kubernetes
- Because things are migrating, need to be able to route traffic between both configurations
- Slow start with small services
- Then all at once
  - with big services

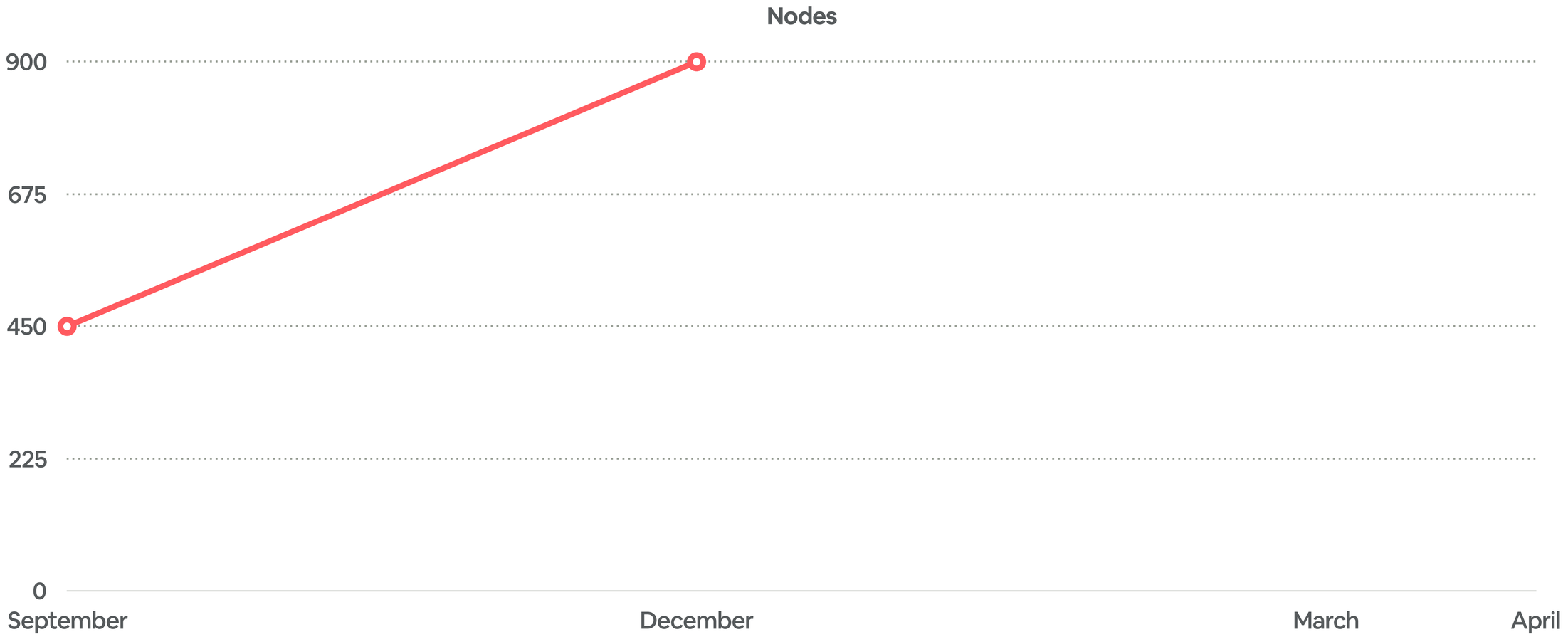
What are our K8s scaling limits?

Growth of Prod Cluster



“We should start thinking about multicluster”

Growth of Prod Cluster



“Uh… what’s the limit on cluster size, again?”

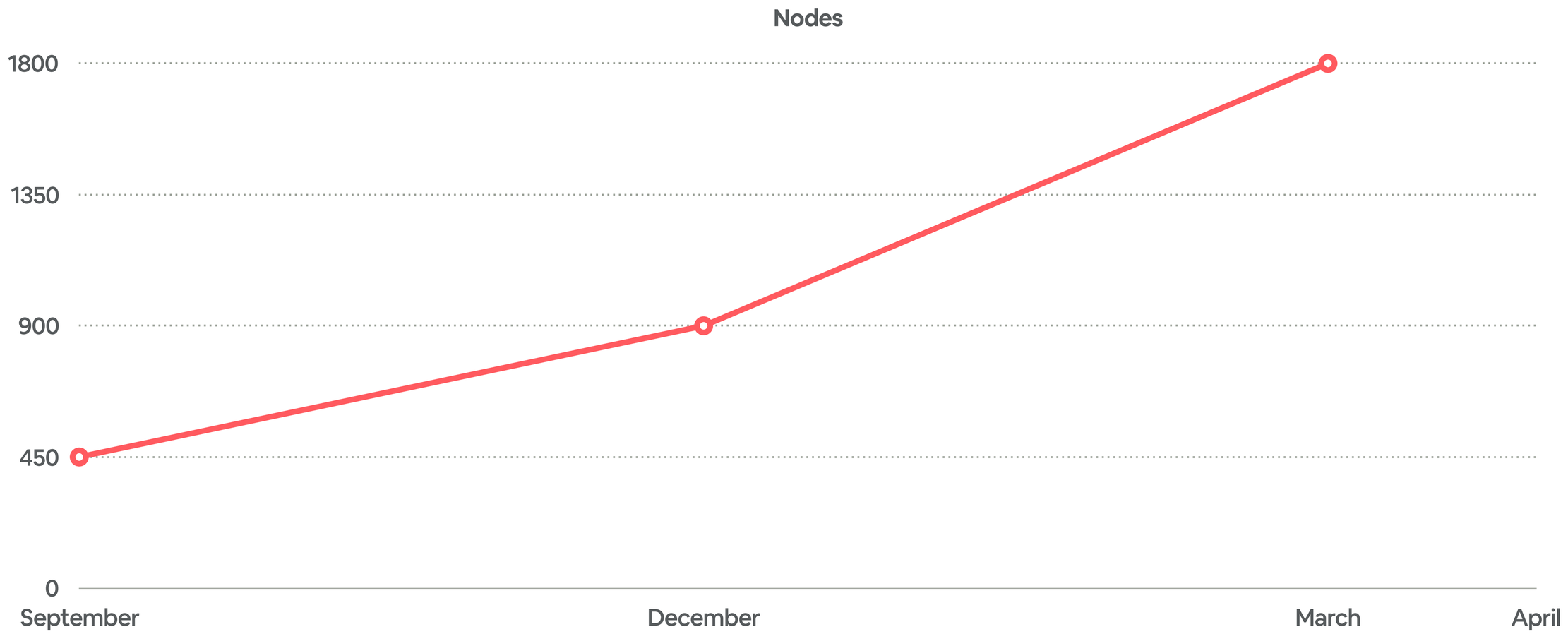
Limits

- Hard limit 5000 nodes
  - https://kubernetes.io/docs/setup/best-practices/cluster-large/
- You can definitely probably do 2500
  - https://openai.com/blog/scaling-kubernetes-to-2500-nodes/
- “Yeah, things get a lot more difficult after 2500”
  - various conversations
- (more recently) 10,000! with a lot of work (great job Alibaba!)
  - https://www.alibabacloud.com/blog/how-does-alibaba-ensure-the-performance-of-system-components-in-a-10000-node-kubernetes-cluster_595469)

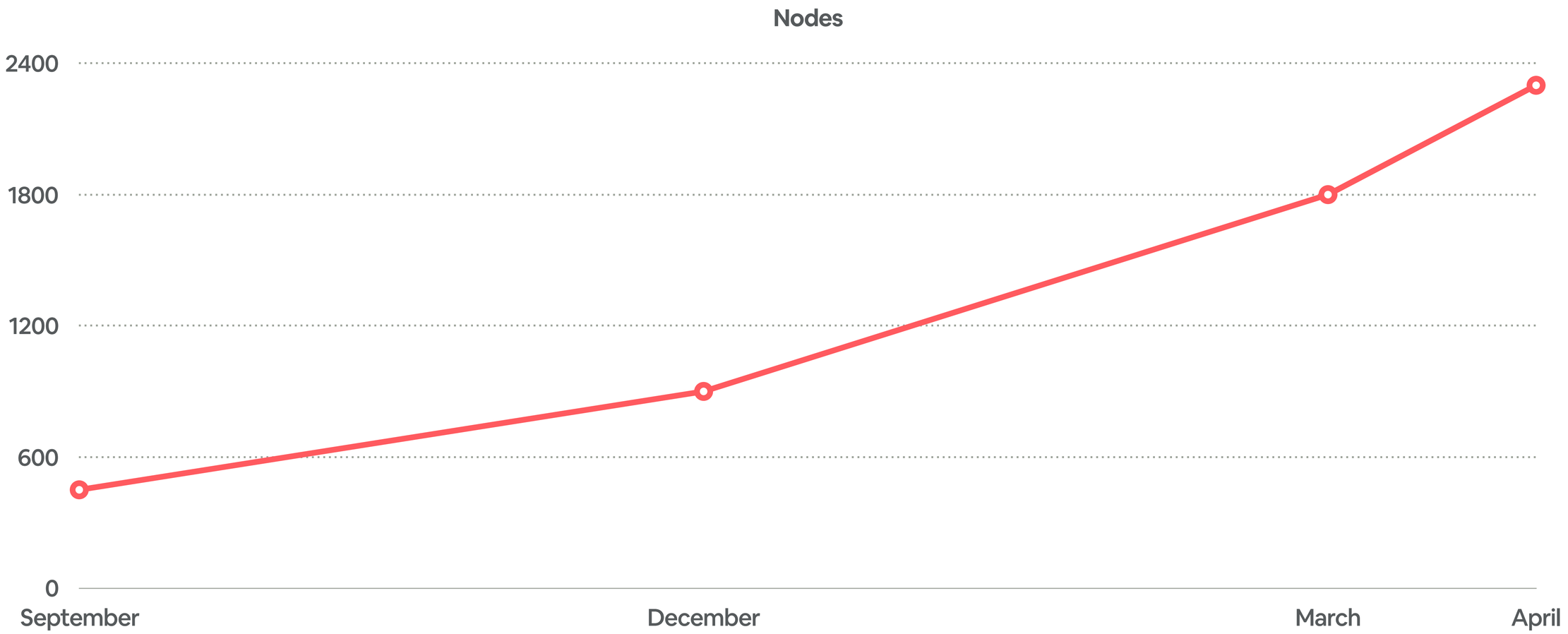
“It would be bad if our etcd were OOMing, right?”

“Because etcd is OOMing”

Growth of Prod Cluster



Growth of Prod Cluster



Scaling to multiple clusters

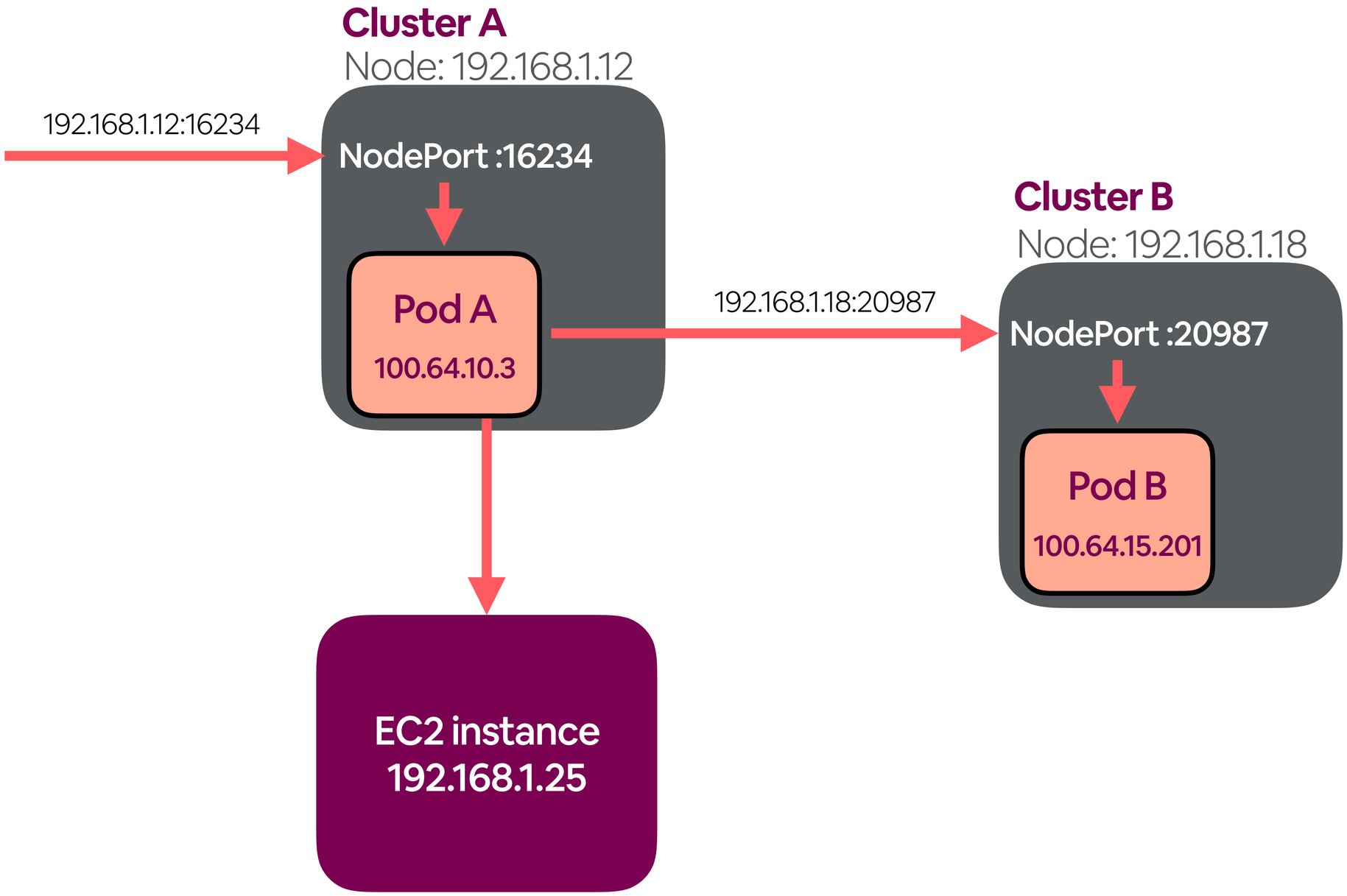
Multicluster considerations

- Are there any placement restrictions on workloads?
- What problem are we solving with multicluster
- A lot can be included here, and things like KubeFed try to account for much of it

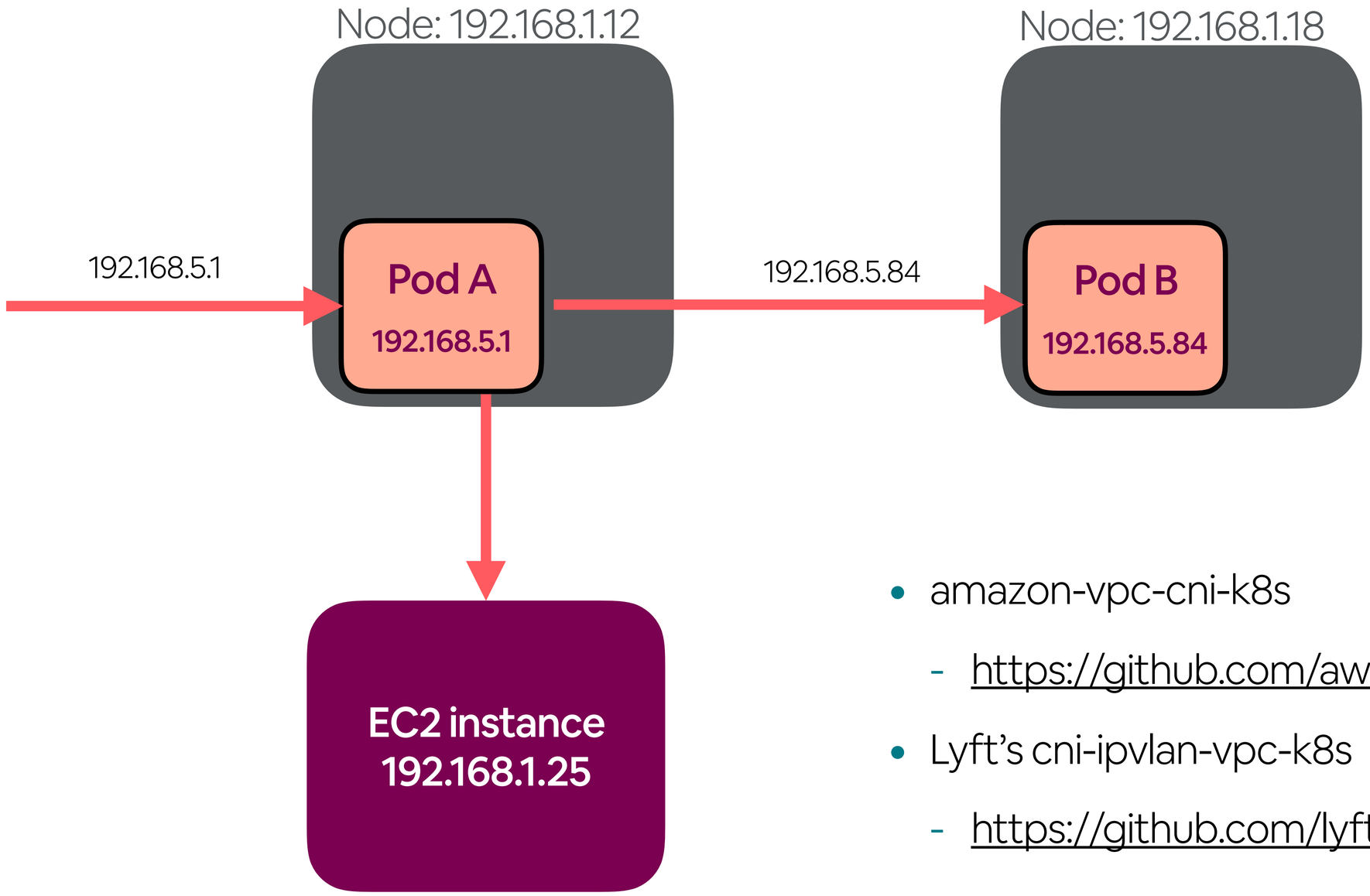
We Got Lucky

- Our Service Mesh (smartstack) and legacy infra meant we had no colocation requirements
- Workloads can be randomly assigned to any cluster
- Clusters are just pools of compute and memory
- At service creation time, we choose a cluster and bind it in the service config

Service Mesh (smartstack)



Equivalent to various VPC CNIs



- amazon-vpc-cni-k8s
  - https://github.com/aws/amazon-vpc-cni-k8s
- Lyft's cni-ipvlan-vpc-k8s
  - https://github.com/lyft/cni-ipvlan-vpc-k8s

Multicluster

- Randomly assign workloads to clusters at creation time
- Grow clusters to a certain size (~400 nodes) before closing off to new services
- Allows services already on the cluster to scale without getting close to size limits
- Some special case clusters for things like test, dev, special security groups, machine learning, etc.

How we manage our clusters

Multi = Many

Cluster = Cluster

- The key thing with multicluster is that you’re going to need to make a lot of clusters
- And keep them consistent

Other Solutions

- Kops, kubeadm, things of that nature
- Great tools if they work for your situation
- For us, we needed to play nicely with our existing infra/ tooling

What is a cluster?

What is a cluster?

- Nodes
- Machine config
- Etcd
- Control Plane config
- Cloud Provider objects (IAM roles, DNS, etc.)
- Certificates
  - Genius of K8s is it reduces all problems to either PKI or networking
- For these we generate chef and terraform code into the appropriate repos.

What is a cluster?

THERE'S MORE

- CNI
- DNS
- metrics-server
- filebeat
- RBAC
- All the cluster services that show up in tutorials as

```
kubectl apply -f https://raw.githubusercontent.com…
```

One Deployable Unit

- Cluster services are like any other workloads we run
- Want to preserve standard development practices
- All cluster services deployed as a single unit
- Avoids missing services from cluster, also helps to avoid version drift
- Simple metric for inclusion: Does this need to exist in every cluster that needs it?
- Refer to this as kube-system

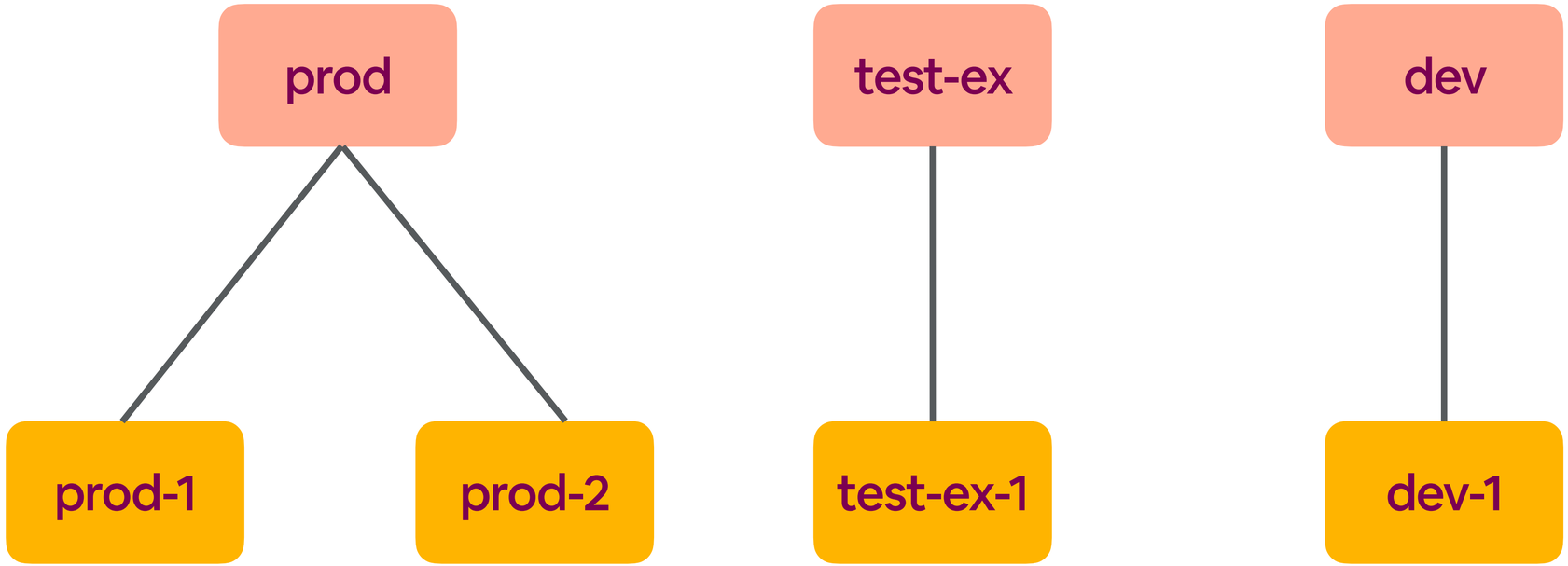
Deploy Process

- Mix of things
- Components written as helm charts, pulled in via an “umbrella chart” per cluster, templated into a single manifest
- Various AWS configuration
- Applications deployed via kube-gen, our in house framework
- Components can fail, aim for predictability, which usually means trying to deploy as fully as possible
- Deploys are < 10 minutes

Organization

Cluster Types

Clusters



Clusters and Cluster Types

- Cluster Types are like classes, Clusters are like instances
- Clusters exist for horizontal scaling
- From a workload's perspective, it should not care nor be able to discern which cluster of a cluster type it is running on

Cluster Launch

- Generate chef, terraform, and kube-system
- Get those PRs approved and merged
- Launch etcd and kubernetes nodes
- Deploy kube-system
- Cluster is ready to use
- Process takes about an hour, much of which is code review
  - Stopped optimizing beyond this point

Dealing with change

Making change easy means that more change will happen

What we've learned

- Originally generators were basically one shot
- Some consistency was maintained by cluster-type sharing
- Changes happened much more often than expected
- And more dramatically
- Second version fully regenerates outputs from input configs for every cluster type/cluster

Some things are easy to change

- Adding a new cluster service, changing its manifest
- Upgrading Kubernetes
- Adding new IAM permissions
- Refactoring machine configuration
- Most stuff

Other things are really, really hard to change

- Networking
  - Type of CNI plugin
  - CIDRs
- Naming conventions
  - “Now we format IAM roles like…”
- Certificates
  - Invalid service account tokens

Cluster Config

- Has *fixed variables* to preserve things that can't be easily updated in place
- Allows for changes to be made *going forward* for new clusters

```
kind: Cluster
name: prod-1
clusterType: prod
status:
  assignable: false
  etcdClusterState: existing
  masterInitialized: true
# Should be little need to ever override these,
these are preserved to allow
# configuration and fix the values independent
of future change
variables:
  clusterNameCamel: Prod1
  masterURL: https://prod-1.kubernetes.foo
  dashboardURL: https://kube-dashboard-
prod-1.foo
  masterChefRole: kubernetes-master-prod-1
  nodeCidrMaskSize: 24
  …
```

Supporting the Unsupported

- Not all use cases are supported (yet)
- Users can drop down and modify the output directly
- Full power of chef, terraform, or whatever
- A standard comment prevents changes from being overwritten

```
# [ ] Check box to prevent automatic overwrites
on regenerate
# Generated by kube-system (lib/kube-system/
types/output/helm_chart.rb)

or

# [x] Check box to prevent automatic overwrites
on regenerate
# Generators do not support per-cluster value
overrides
```

Ongoing process, still learning what’s needed

Where we are

- **22** Cluster Types
- **36** Clusters
- **7000+** nodes

Thanks!

- Engineering blog at medium.com/airbnb-engineering
- Jobs at careers.airbnb.com/