



KubeCon



CloudNativeCon

North America 2017

Scaling to 5000+ K8s Deployments

Nicole Hubbard, Architect, *WP Engine*

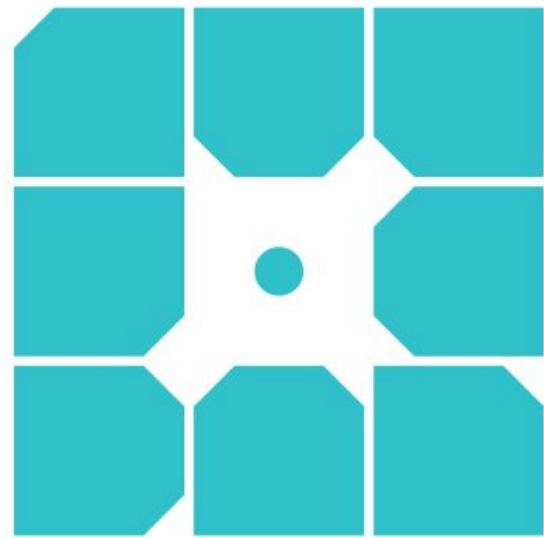


Nicole Hubbard

- Devops Architect
- @nicolerenee3810
- Sparkly Devops Princess



Scaling to 5000 K8s Deployments



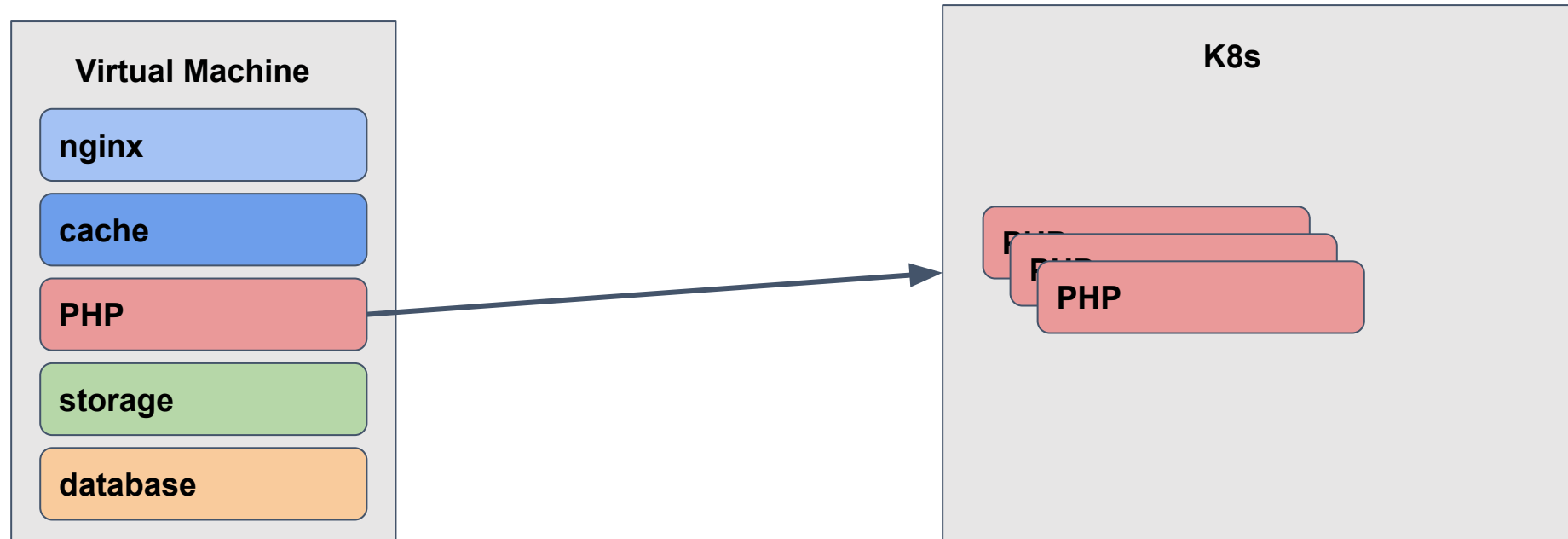
WPengine

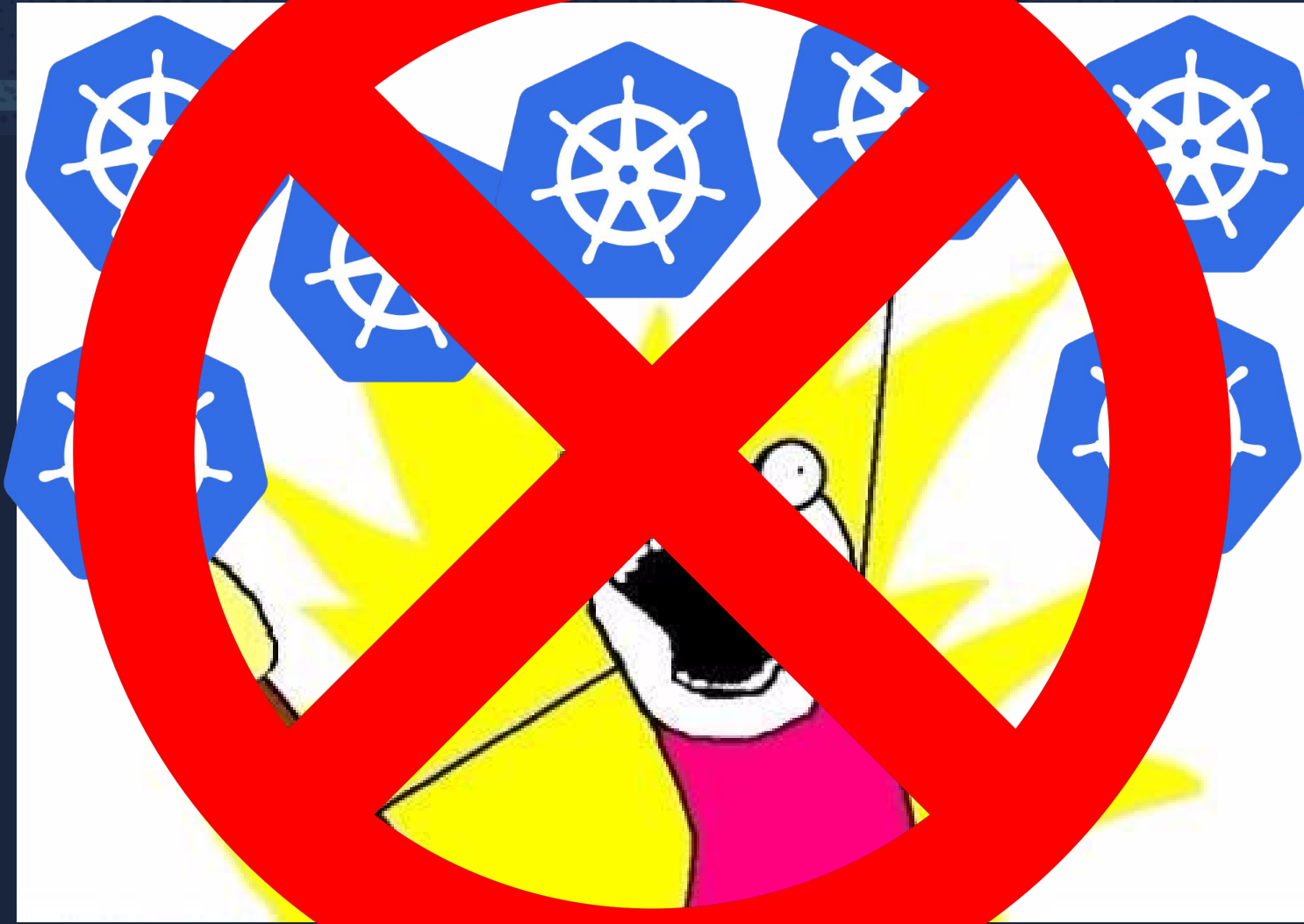
- Over 300k WordPress installs
- 5% of the online world visits at least one site we host each day
- Transitioning from existing VM platform to K8s

WordPress

- Open Source project started in 2003
- Runs on a traditional LAMP stack
- Requires shared file system
- Users install Plugins and modify code
- Requires access to the filesystem

What are we doing





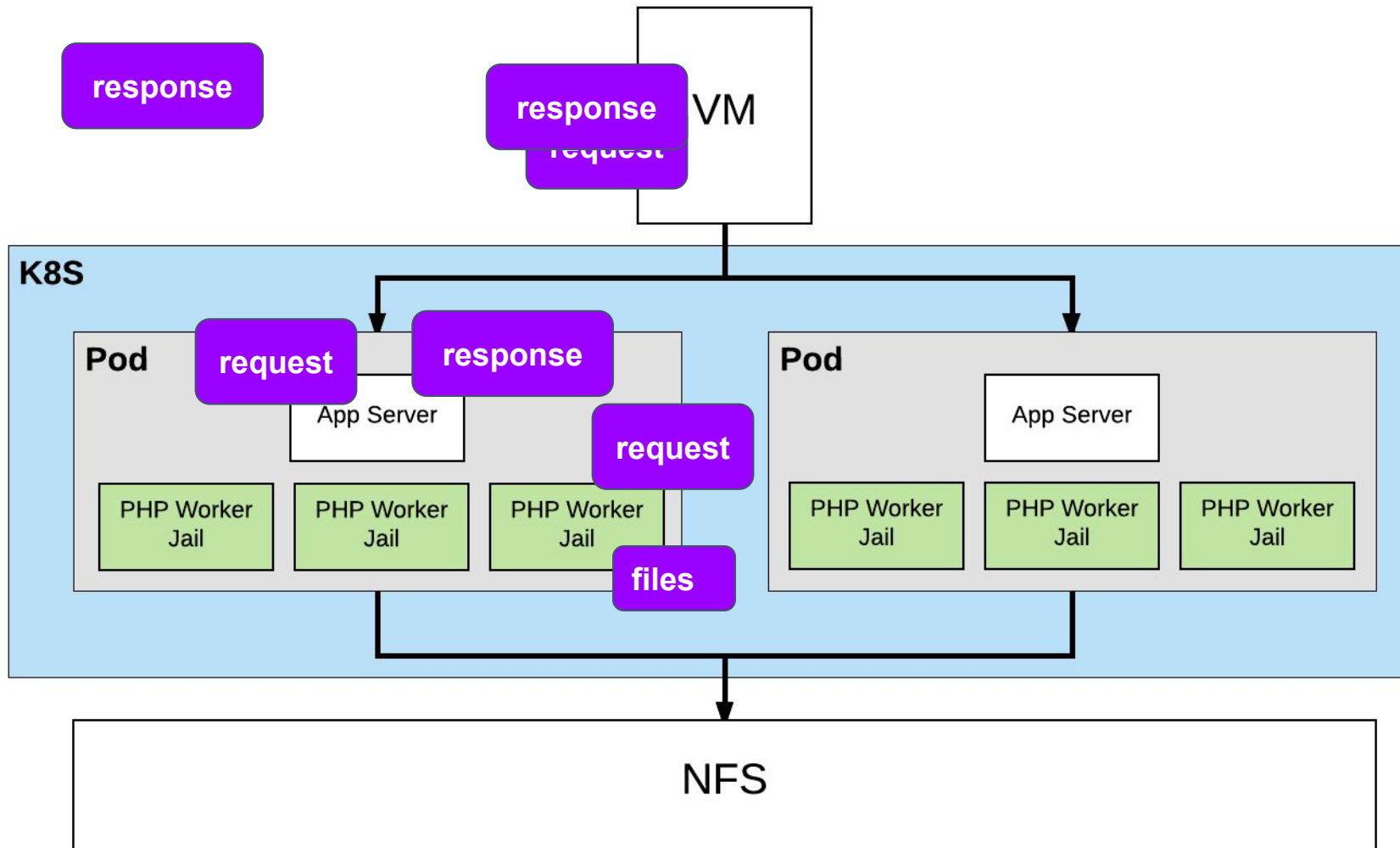
Kubernetes 1.8 Limits

- 150,000 pods per cluster
- 100 pods per Node
- Website resource utilization is highly variable
- 100 pods per node isn't dense enough

Custom PHP Application Server

- Written in Go
- Runs PHP-FPM workers in namespace jails
- Just in time mounts site content into the jail
- Able to deploy an instance per existing VM

Custom PHP Application Server



300,000 deployments → 6,000 deployments

Helm

- Helm requires 3 pieces of information
 - What do you want me to do? (chart)
 - What are the settings? (values)
 - Where do I do it? (tiller)

Deployment Problems

- 6000+ PHP worker deployments
- Helm is our preferred K8s deployment method
- Over 6,000 unique values files to try and maintain
- 7 regional K8s clusters

How?

Operators

- Examples of existing operators
 - Elasticsearch
 - Kafka
 - Redis
 - etcd
 - Prometheus

Custom Resources

- Allows you to extend the K8s API
- Represents any object you want in K8s
- Store any data you want in the object
- Added in K8s 1.7

```
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  name: characters.stable.nicolereene.io
spec:
  scope: Namespaced
  group: stable.nicolereene.io
  version: v1
  names:
    kind: Character
    plural: characters
    singular: character
```


apiVersion: stable.nicolereene.io/v1

kind: Character

metadata:

name: thing1

spec:

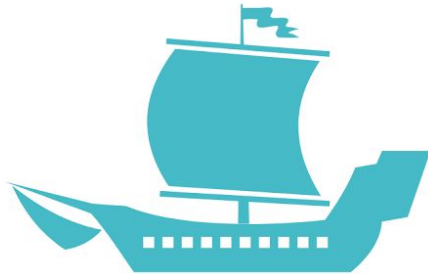
name: Thing 1

from: Cat in the Hat

by: Dr. Seuss

```
~/talks/2017-kubernetes vi cr_nemo.yml
~/talks/2017-kubernetes kubectl apply -f cr_nemo.yml
character "nemo" configured
~/talks/2017-kubernetes kubectl describe characters nemo
Name:          nemo
Namespace:     default
Labels:        <none>
Annotations:   kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"stable.nicolereene.io/v1","kind":"Character","metadata":{"annotations":{},"name":"nemo","namespace":"default"},"spec":{"by":"Pixar","from":
...
API Version:  stable.nicolereene.io/v1
Kind:         Character
Metadata:
  Cluster Name:
  Creation Timestamp:      2017-11-17T04:40:01Z
  Deletion Grace Period Seconds: <nil>
  Deletion Timestamp:      <nil>
  Generation:              0
  Resource Version:        425480
  Self Link:                /apis/stable.nicolereene.io/v1/namespaces/default/characters/nemo
  UID:                      5f5a8e5f-cb51-11e7-9700-caa21752e25e
Spec:
  By:      Pixar
  From:    Finding Nemo
  Name:    Nemo
Events:    <none>
~/talks/2017-kubernetes kubectl get characters:stable.
```

Lostrómos



LOSTRÓMOS

λοστρόμος (lostrómos) = boatswain

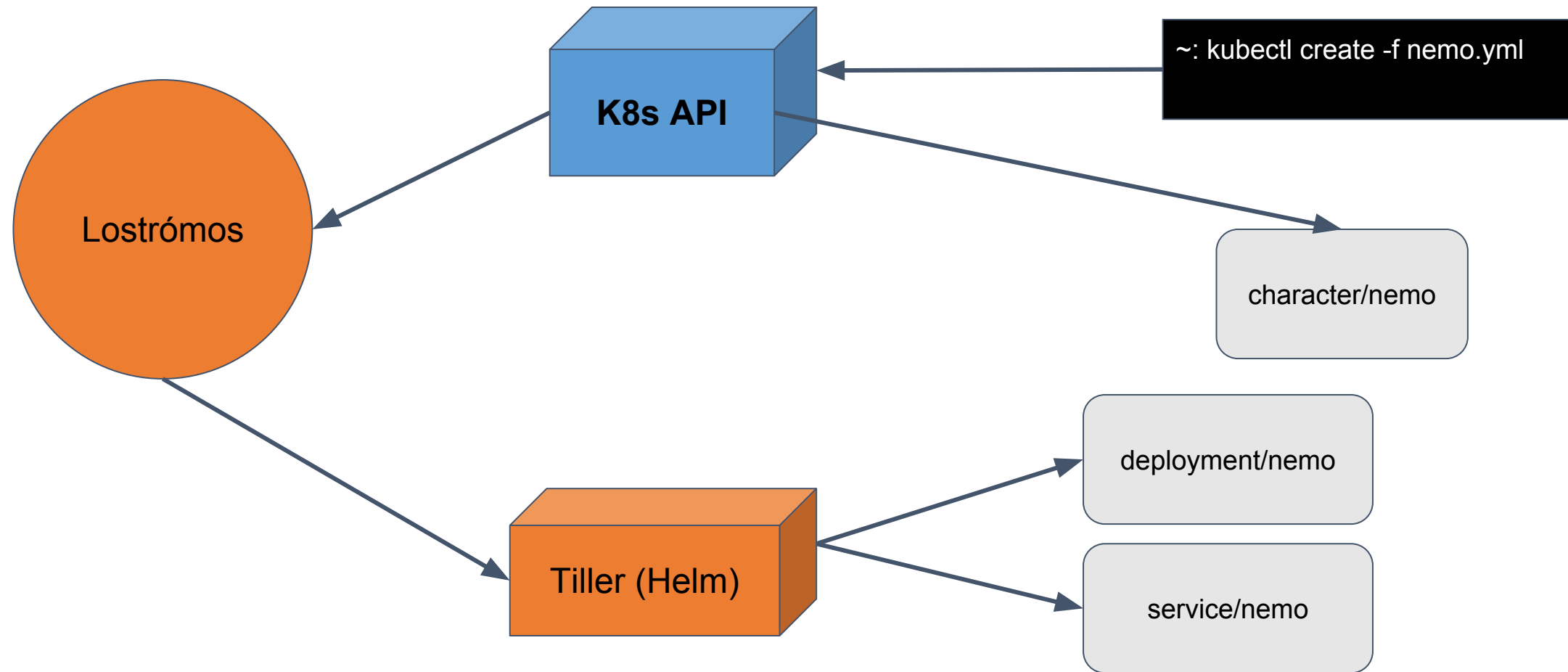
boatswain (n): a warrant officer on a warship, or a petty officer on a merchant vessel, in charge of rigging, anchors, cables, etc.

github.com/wpengine/lostromos

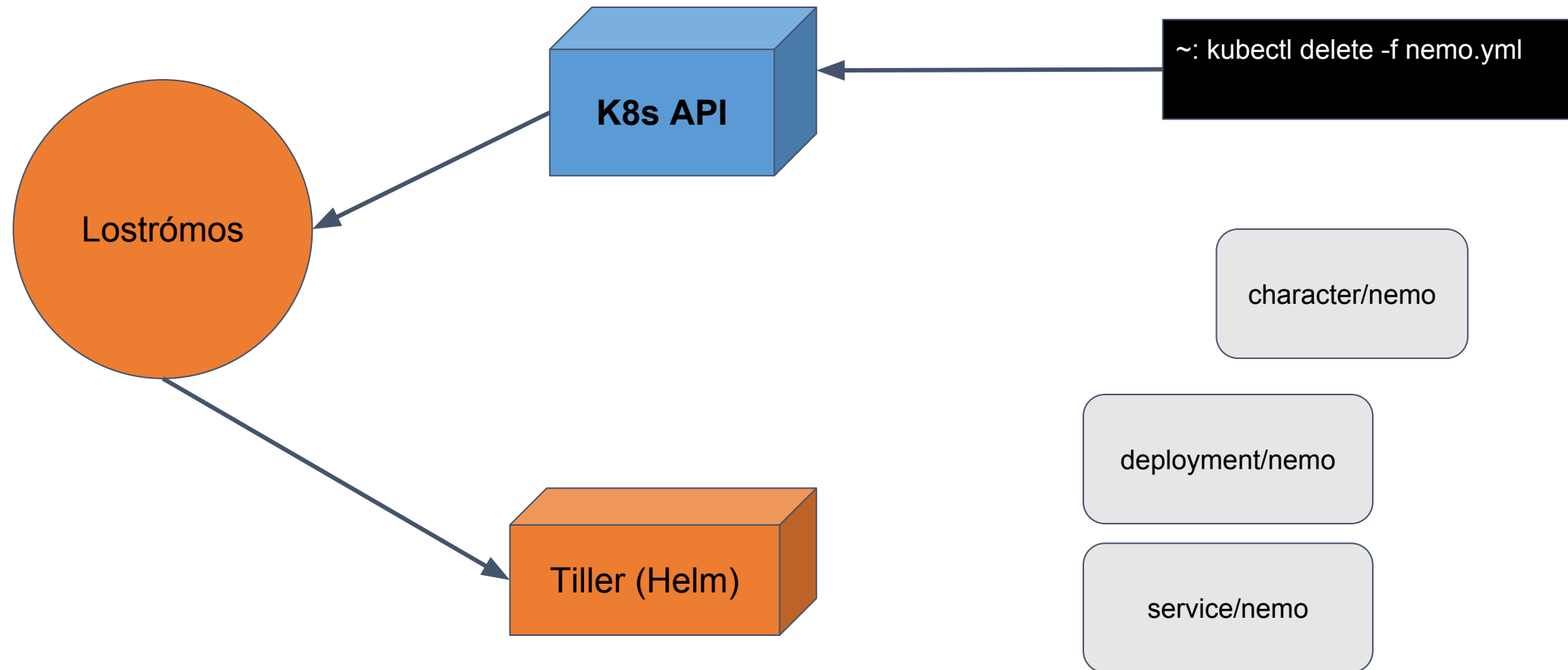
Lostrómos

- Monitors Custom Resources
- Deploys helm chart for every Custom Resource
- Handles changes on the Custom resource
 - Creations
 - Updates
 - Deletions

Lostrómos



Lostrómos



```
2017-11-17T01:11:12.517-0600 INFO version/version.go:38 version info {"version": "v1.8.0-alpha.1", "gitCommit": "71ac158", "buildTime": "2017-11-06_18:14:50PM"}
2017-11-17T01:11:12.526-0600 INFO cmd/start.go:132 using helm controller for deployment {"controller": "helm", "helmChart": "demo-chart", "helmNamespace": "demo", "helmReleasePrefix": "lostronom", "helmRelease": "127.0.0.1:44134"}
2017-11-17T01:11:12.541-0600 INFO helmctl/helm.go:62 resource added {"controller": "helm", "resource": "thing2"}
2017-11-17T01:11:12.554-0600 INFO helmctl/helm.go:62 resource added {"controller": "helm", "resource": "thing1"}

```

root@kubem: ~

```
Last login: Fri Nov 17 01:24:15 on ttys004
root@kubem:~# helm ls
NAME                REVISION      UPDATED                               STATUS          CHART              NAMESPACE
lostronom-thing1    1             Fri Nov 17 01:31:28 2017        DEPLOYED        kubecore-demo-0.1.0  demo
lostronom-thing2    1             Fri Nov 17 01:31:28 2017        DEPLOYED        kubecore-demo-0.1.0  demo
root@kubem:~# kubectl -n demo get services
NAME                                TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
lostronom-thing1-kubecore-demo     NodePort    10.0.0.19        <none>            80:30881/TCP     35s
lostronom-thing2-kubecore-demo     NodePort    10.0.0.81        <none>            80:32166/TCP     35s
root@kubem:~# kubectl -n demo get deployments
NAME                                DESIRED      CURRENT    UP-TO-DATE    AVAILABLE   AGE
lostronom-thing1-kubecore-demo     1            1          1              1           46s
lostronom-thing2-kubecore-demo     1            1          1              1           46s
root@kubem:~# minikube service -n demo lostronom-thing1-kubecore-demo
Opening kubernetes service demo/lostronom-thing1-kubecore-demo in default browser...
root@kubem:~# curl http://192.168.64.11:30881/
Thing 1 first appeared in Cat in the Hat created by Dr. Seuss.
root@kubem:~# kubectl apply -f cr-nano
```

Challenges

- How fast should deploys be?
- How do we deploy new versions?
- Monitoring
- Reconciliation
 - What if we miss an event because we are offline
 - What if someone changes something

Examples of Other Use Cases

- Databases for your Application
- Monitoring Agent for CloudSQL/RDS

- lostromos/crwatcher
 - Update DNS with your provider via CR
 - Create cloud resources via CR

Future

Some ideas we have for the future

- Support for watching resources beyond custom resources
- Support additional deployment mechanisms

Other ideas are welcomed and PRs are happily accepted!

Questions?

github.com/wpengine/lostromos