





Managing Kubernetes:

Day 2 Operations

Craig Tracey, Staff Field Engineer

 [craigtracey](#)

 [craig_tracey](#)

 theothercraig@heptio.com

About



NETEZZA



CARBONITE

HubSpot

bluebox
AN IBM COMPANY

 heptio



Brendan Burns & Craig Tracey

Day 2 Operations

Which would you prefer?



Proactive



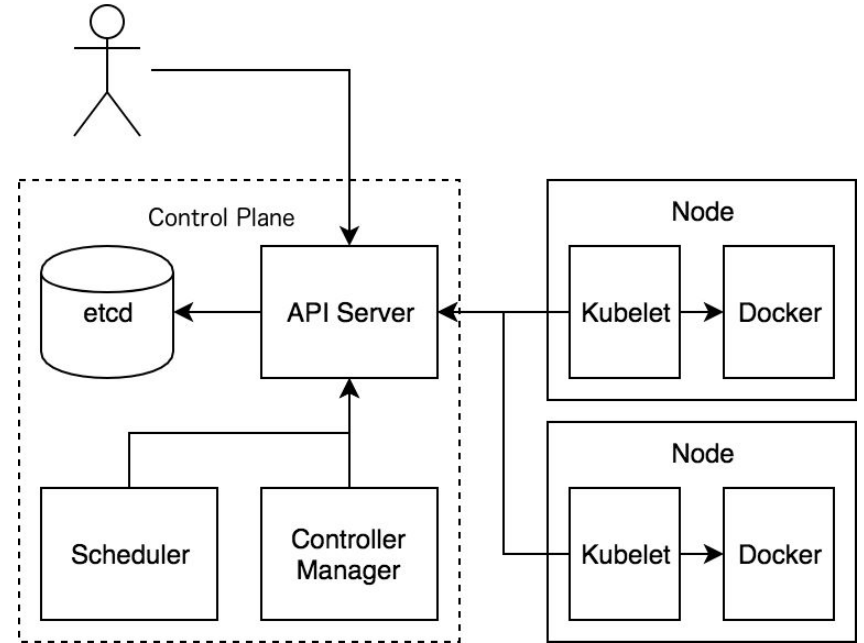
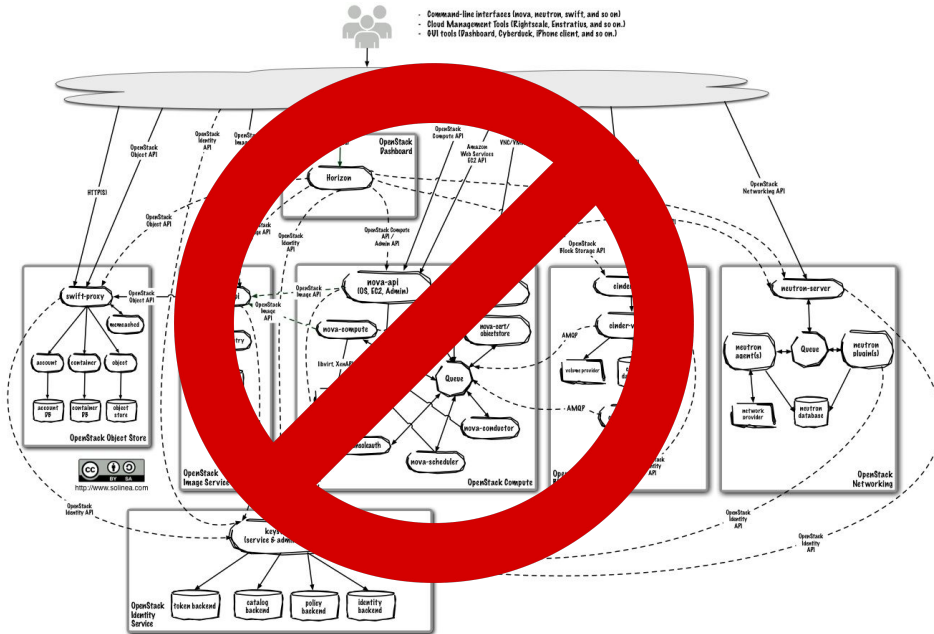
Reactive



Successful Kubernetes
deployments are just
as much a cultural challenge
as they are technical.

simplicity

noun | sim-PLIC-i-ty | \sim-'pli-sə-tē, -'pli-stē\



<https://blog.heptio.com/core-kubernetes-jazz-improv-over-orchestration-a7903ea92ca>

User Experience



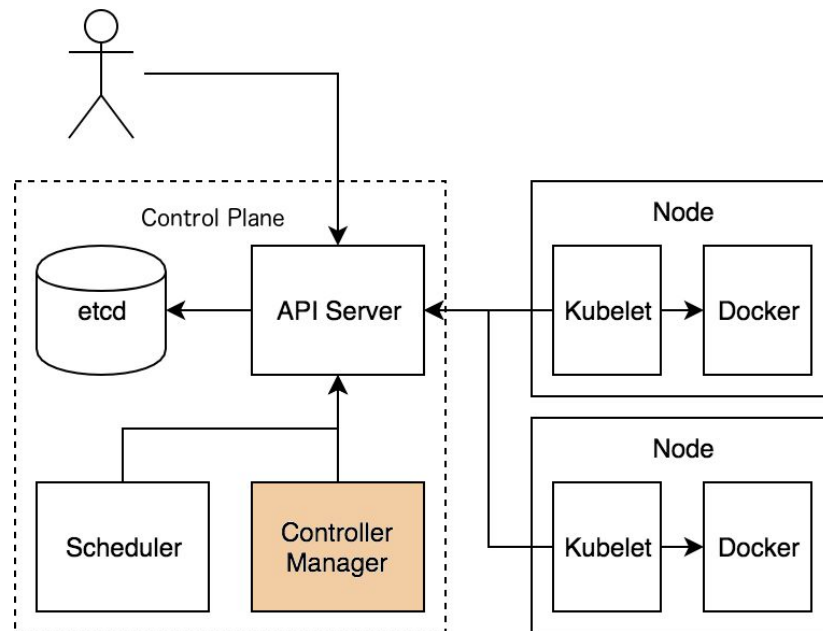
- Don't boil the ocean
- If you have great tools, use them
- Optics matter

Tactically

Platform

API Server

Controllers



Platform

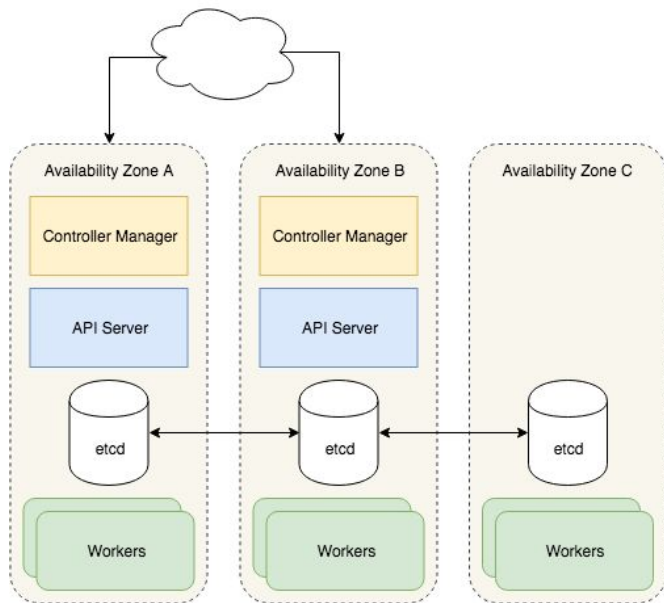
- Day 1 concerns
- Installation matters (today)
- Hardware tuning
- Component life cycles
- Sonobuoy
- Ark

```
// We disable the insecure port from 1.6 onwards
if b.IsKubernetesGTE("1.6") {
    c.InsecurePort = 0
    glog.V(4).Infof("Enabling apiserver insecure port, for healthchecks (issue #43784)")
    c.InsecurePort = 8080
} else {
    c.InsecurePort = 8080
}
```

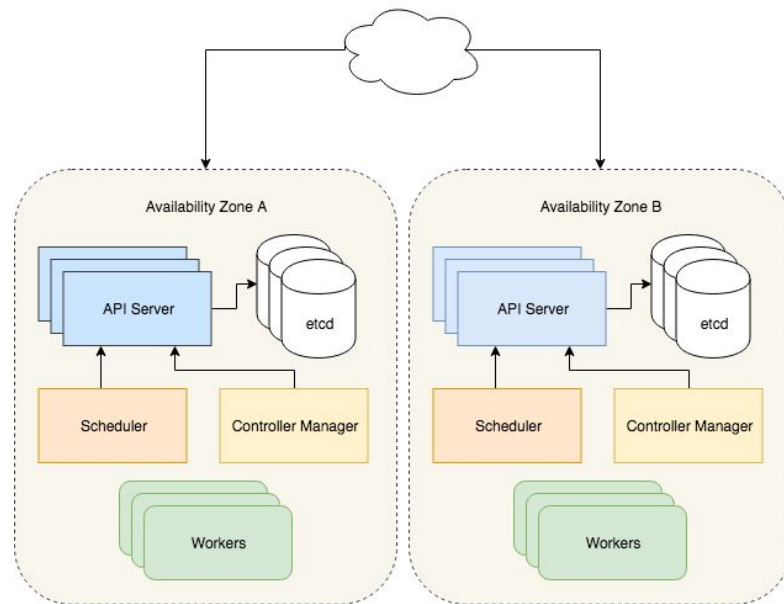


Platform: Availability

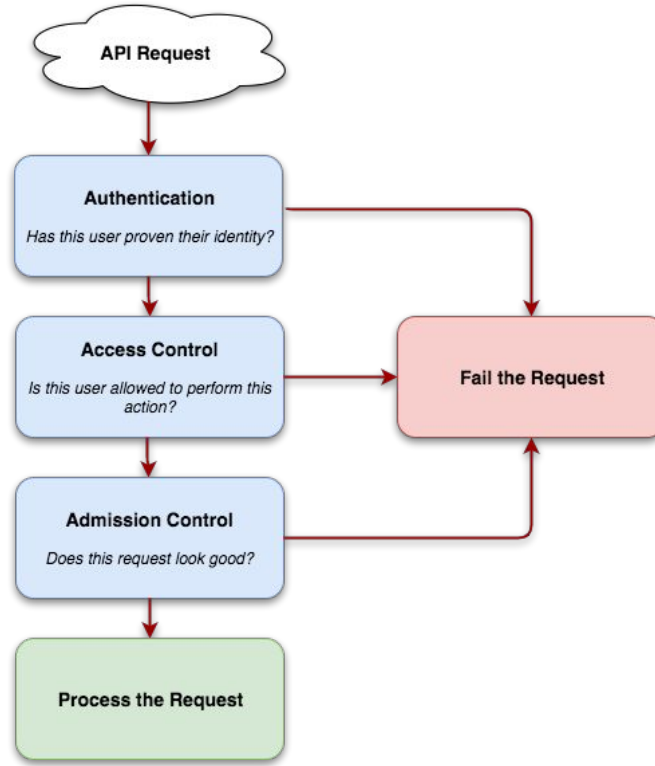
Single Cluster, Multi-AZ



Cluster Per AZ



API Request Flow



API: Authorization



- RBAC!
- UserInfo
- Secrets

```
{
  "iss": "https://auth.example.com",
  "sub": "Ch5hdXRoMHwMTYzOTgzZTdjan2EyNWQxMDViNjESBWF1N2Q2",
  "aud": "dDb1g7x07dksluG6Op976jC7TjUZDCDz",
  "exp": 1517266346,
  "iat": 1517179946,
  "at_hash": "OjgZQ0vauibNVcXP52CtoQ",
  "username": "jane",
  "email": "jane@example.com",
  "email_verified": true,
  "groups": [
    "qa",
    "infrastructure"
  ]
}
```

```
75 // UserInfo holds the information about the user needed to implement the
76 // user.Info interface.
77 type UserInfo struct {
78     // The name that uniquely identifies this user among all active users.
79     Username string
80     // A unique value that identifies this user across time. If this user is
81     // deleted and another user by the same name is added, they will have
82     // different UIDs.
83     UID string
84     // The names of groups this user is a part of.
85     Groups []string
86     // Any additional information provided by the authenticator.
87     Extra map[string]ExtraValue
88 }
```

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  namespace: default
  name: pod-reader
rules:
- apiGroups: ["" ] # "" indicates the core API group
  resources: ["pods"]
  verbs: ["get", "watch", "list"]
```

```
# This role binding allows "jane" to read pods in the "default" namespace.
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: read-pods
  namespace: default
subjects:
- kind: User
  name: jane # Name is case sensitive
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: pod-reader
  apiGroup: rbac.authorization.k8s.io
```

API: Admission Control

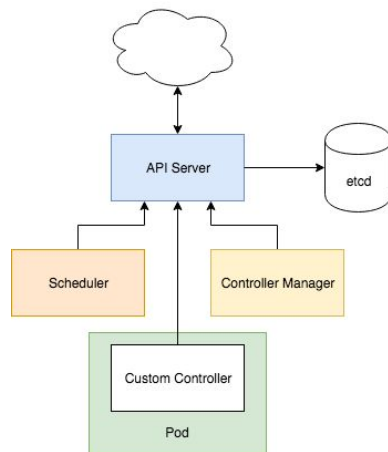


- Business Rules
- Quotas
- Pod Security Policies
- Dynamic Admission Control

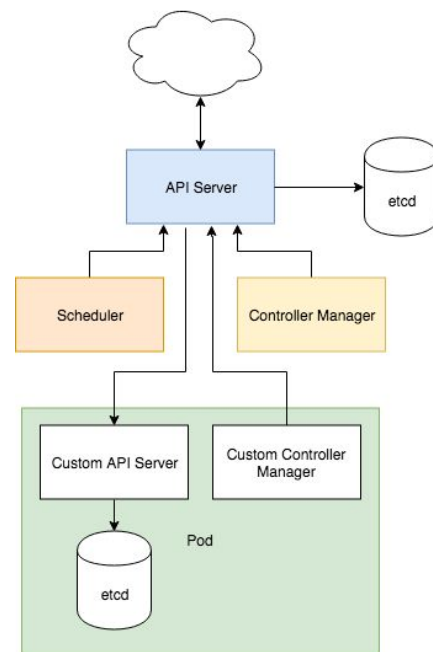
Custom Controllers

- Kubernetes is not magic
- Business logic
- Any language
- Understand limitations

Custom Controllers / CRD



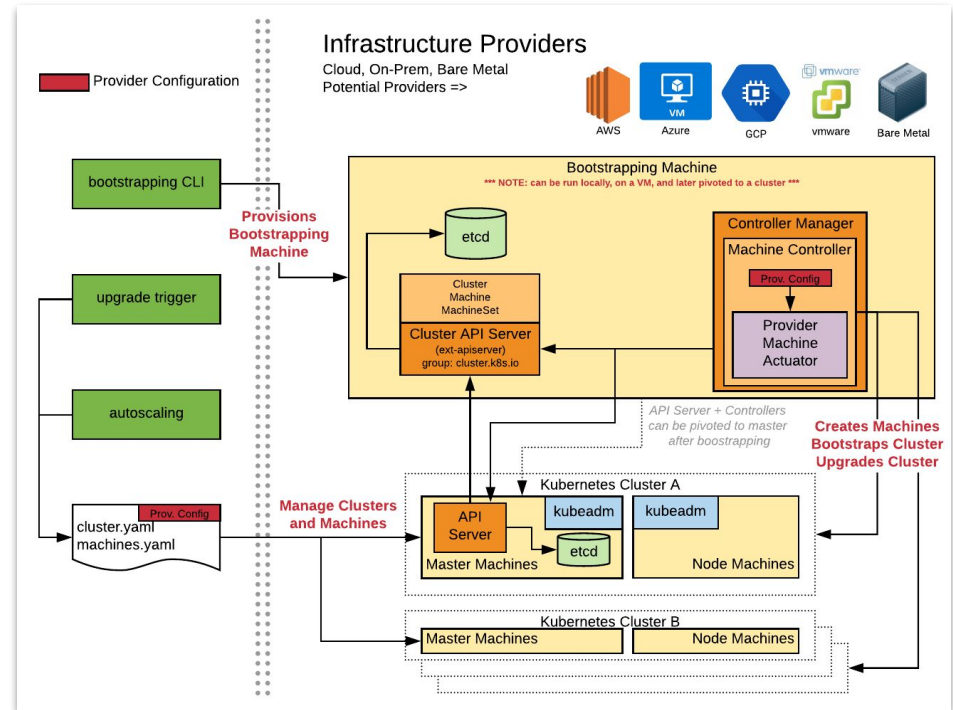
Aggregate API



(not-so-distant) Future: Cluster API



- Familiar Interface
- Commoditize Deployment
- Standardize Operations
- Registry



<https://github.com/kubernetes-sigs/cluster-api>

The Voltron Moment

Bringing it all together

- Simplicity
- User experience
- Platform
- API Request Flow
- Custom Controllers



References



<https://github.com/heptio/ark>

<https://github.com/heptio/sonobuoy>

<https://github.com/heptiolabs/gangway>

<https://github.com/coreos/dex>



Thanks!