KubeCon

North America 2017

Stupid Kubectl Tricks

Jordan Liggitt, *Red Hat*

1. Personality Test

1. Personality Test

“kubectl”

1. Personality Test

kube·con·trol

engineer control freak

kube control to Major Tom

1. Personality Test

kube·cut·tle

scary operations expert

8 arms. 2 tentacles. 0 pets.

1. Personality Test

kube·ee·cud·dle

open-source hippie

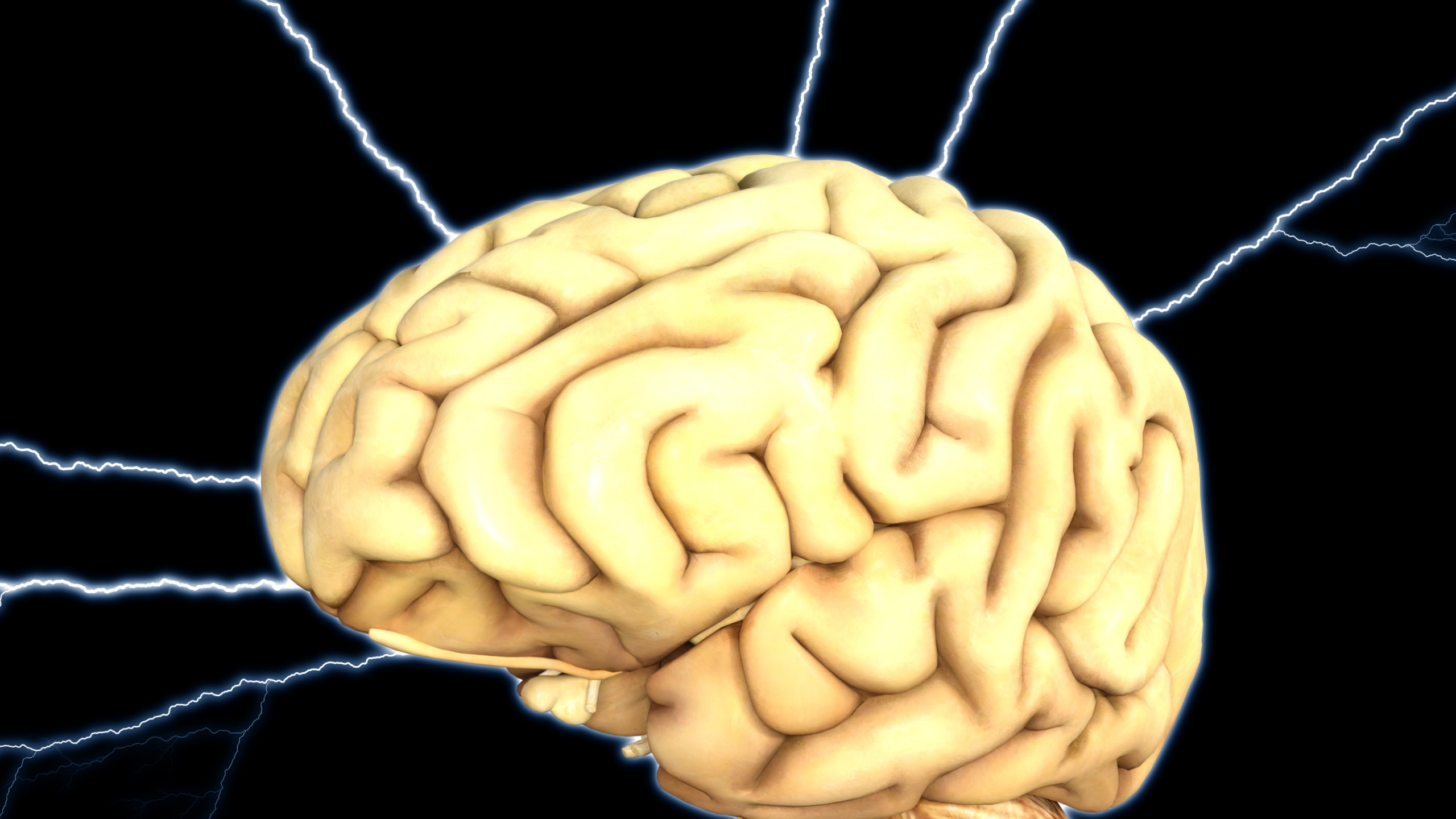
live. love. linux.

1. Personality Test

<anything else>

special snowflake

just like everyone else



2. Autocomplete

```
$ source <(kubectl completion bash)
```

2. Autocomplete

```
$ kubectl
alpha           cluster-info    describe
annotate        completion      drain
api-versions    config          edit
apply           convert         exec
attach          cordon          explain
```

2. Autocomplete

```
$ kubectl get
certificatesigningrequest  ingress
configmap                  networkpolicy
cronjob                    persistentvolume
daemonset                  persistentvolumeclaim
endpoints                  poddisruptionbudget
horizontalpodautoscaler    podtemplate
```

2. Autocomplete

```
$ kubectl get horizontalpodautoscaler --
--all-namespaces
--allow-missing-template-keys
--alsologtostderr
--as-group=
--as=
```

3. Trace logging

```
$ kubectl get pods --all-namespaces

GET /api/v1/pods
```

3. Trace logging

```
$ kubectl run pi --schedule="0/5 * * * ?" \
    --image=perl --restart=OnFailure \
    -- perl -Mbignum=bpi -wle 'print bpi(2000)'
```

3. Trace logging

-v

3. Trace logging

```
$ kubectl run -v=6 pi --schedule="0/5 * * * ?" \
    --image=perl --restart=OnFailure \
    -- perl -Mbignum=bpi -wle 'print bpi(2000)'

POST /apis/batch/v1beta1/namespaces/default/cronjobs
```

3. Trace logging

```
$ kubectl run -v=7 pi --schedule="0/5 * * * ?" \
    --image=perl --restart=OnFailure \
    -- perl -Mbignum=bpi -wle 'print bpi(2000)'
…
Request Headers:
    User-Agent: kubectl/v1.9.0 (darwin/amd64)
    Accept: application/json, */*
    Content-Type: application/json
```

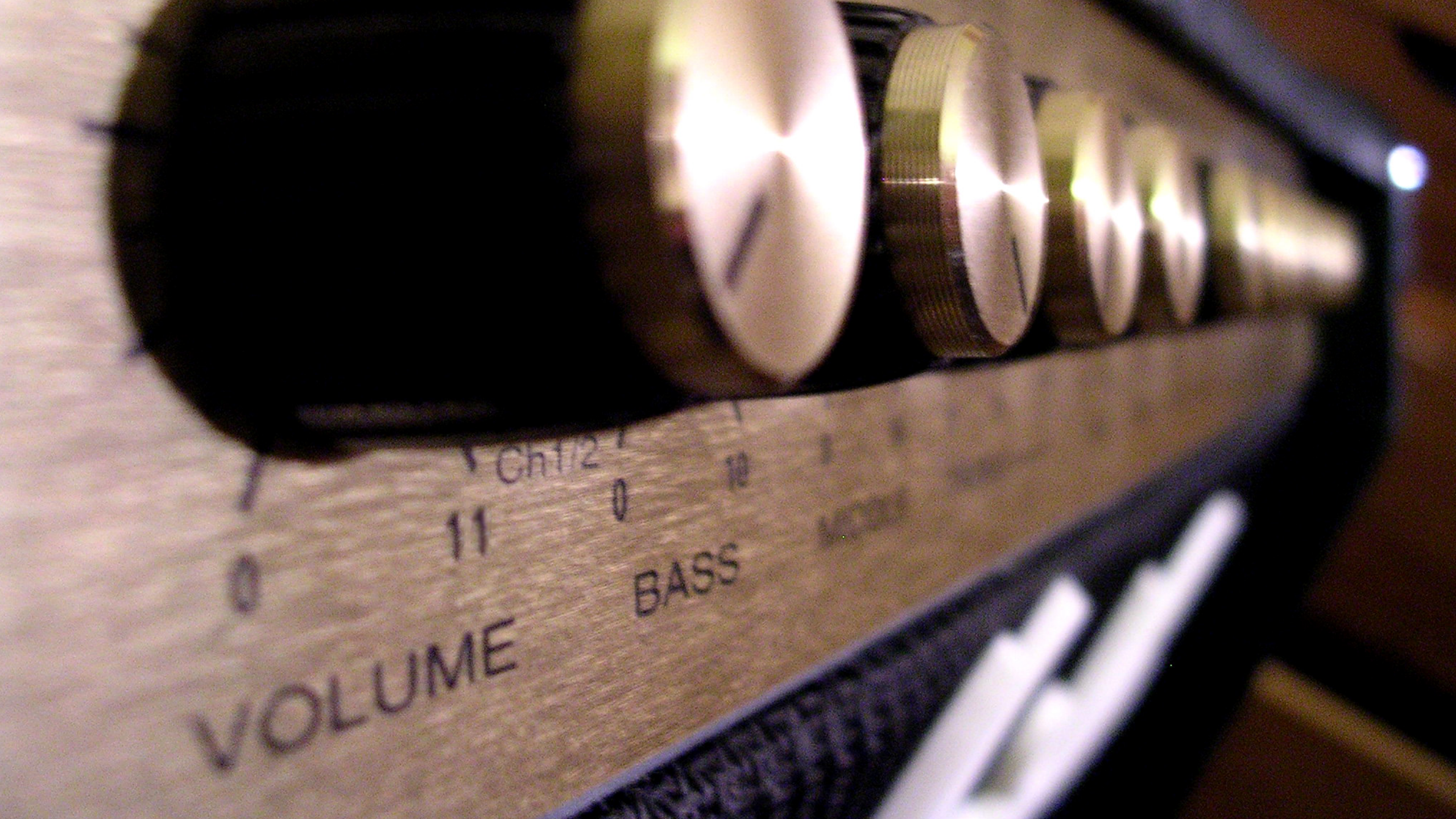
3. Trace logging

```
$ kubectl run -v=8 pi --schedule="0/5 * * * ?" \
    --image=perl --restart=OnFailure \
    -- perl -Mbignum=bpi -wle 'print bpi(2000)'
…
Request Body: {"kind":"CronJob","apiVersion":"batch/
v1beta1","metadata":{"name":"pi",…
…
Response Body: {"kind":"CronJob","apiVersion":"batch/
v1beta1","metadata":{"name":"pi",…
```

3. Trace logging

```
$ kubectl run -v=9 pi --schedule="0/5 * * * ?" \
    --image=perl --restart=OnFailure \
    -- perl -Mbignum=bpi -wle 'print bpi(2000)'
```

```
curl -k -v -XPOST  -H "Accept: application/json, */*" -H
"Content-Type: application/json" -H "User-Agent:
kubectl/v1.9.0 (darwin/amd64) kubernetes/94645c4"
https://localhost:6443/apis/batch/v1beta1/namespaces/
default/cronjobs
```


Ch1/2
11
0
VOLUME
BASS

4. Qualified Resources

```
$ kubectl get deployments/mydep -o yaml

kind: Deployment
apiVersion: ¯\_(ツ)_/¯
metadata:
  name: mydep
  …
```

4. Qualified Resources

```
$ kubectl get <resource>

$ kubectl get <resource>.<group>

$ kubectl get <resource>.<version>.<group>
```

4. Qualified Resources

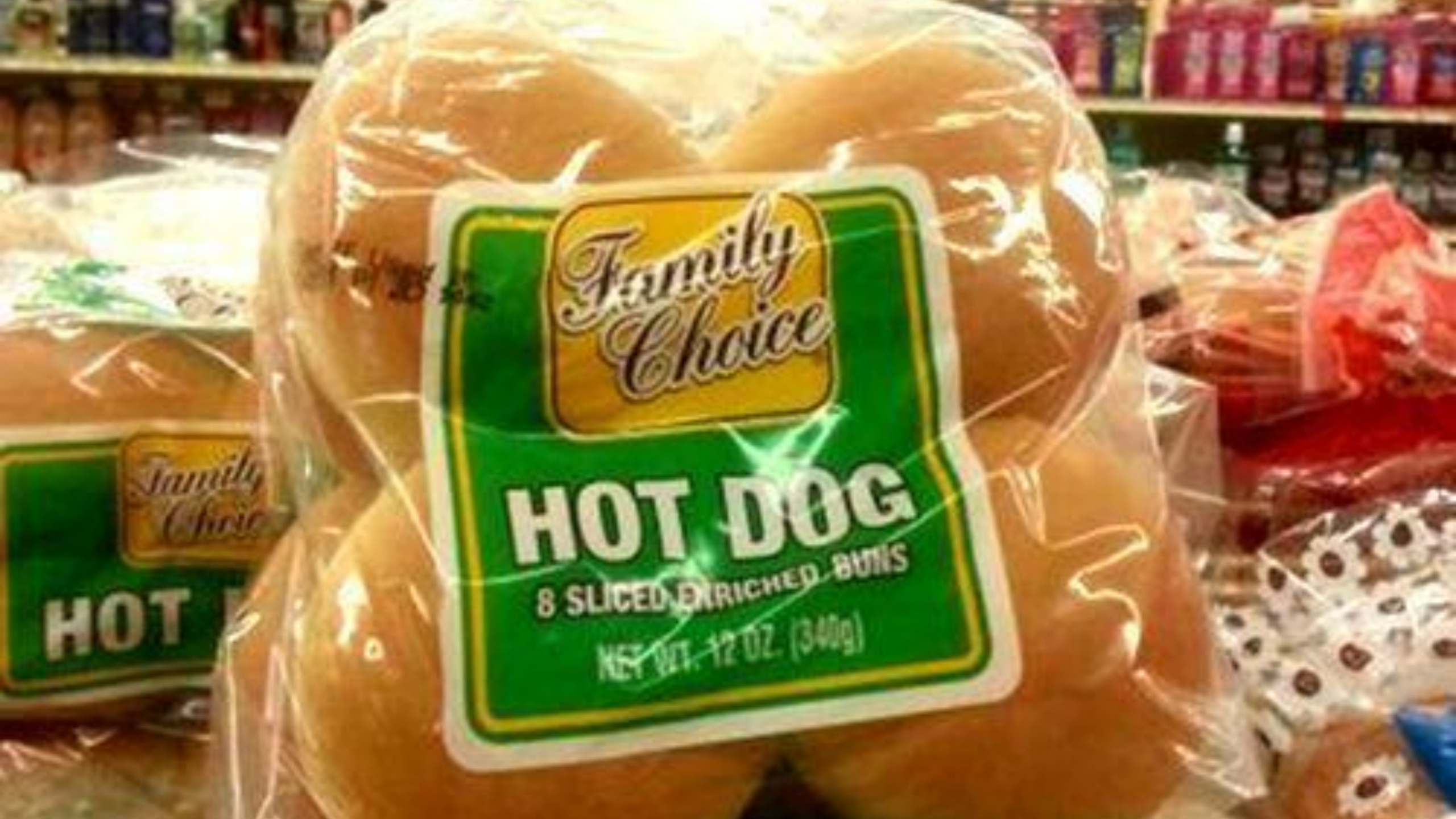
```
$ kubectl get deployments

$ kubectl get deployments.apps

$ kubectl get deployments.v1.apps
```

5. Conversion

```
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: apache
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: apache
  minReplicas: 1
  maxReplicas: 10
  targetCPUUtilizationPercentage: 50
```

5. Conversion

```
$ sed s#autoscaling/v1#autoscaling/v2beta1#
```


Family Choice
HOT DOG
8 SLICED ENRICHED BUNS
NET WT. 12 OZ. (340g)

5. Conversion

```yaml
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: apache
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: apache
  minReplicas: 1
  maxReplicas: 10
  targetCPUUtilizationPercentage: 50
```

```yaml
apiVersion: autoscaling/v2beta1
kind: HorizontalPodAutoscaler
metadata:
  name: apache
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: apache
  minReplicas: 1
  maxReplicas: 10
  metrics:
  - type: Resource
    resource:
      name: cpu
      targetAverageUtilization: 50
```

5. Conversion

```
$ kubectl convert --output-version=autoscaling/v2beta1
```

6. Explain

```
$ kubectl explain <resource> --recursive
```

6. Explain

```
$ kubectl explain hpa --recursive

DESCRIPTION:
  configuration of a horizontal pod autoscaler.
FIELDS:
  spec  <Object>
    maxReplicas    <integer>
    minReplicas    <integer>
    scaleTargetRef <Object>
    …
```

6. Explain

```
$ kubectl explain hpa --recursive \
    --api-version=autoscaling/v2beta1
DESCRIPTION:
  specification of a horizontal pod autoscaler.
FIELDS:
  spec  <Object>
    maxReplicas  <integer>
    minReplicas  <integer>
    metrics      <[]Object>
    …
```

6. Explain

```
$ kubectl explain life

the server doesn't have a resource type "life"
```

7. Piping

```
$ cat greeting
🍗🍗🍗 Happy Thanksgiving!

$ kubectl create configmap greeting --from-file=greeting
configmap "greeting" created
```

7. Piping

```
$ cat greeting
🎄🎄🎄 Merry Christmas!

$ kubectl create configmap greeting --from-file=greeting
Error from server: configmaps "greeting" already exists
```

7. Piping

```
$ kubectl create configmap greeting \
  --from-file=greeting --dry-run --output json
{
  "kind": "ConfigMap",
  "apiVersion": "v1",
  "metadata": {"name": "greeting"},
  "data": {"greeting": "🎄🎄🎄 Merry Christmas!"}
}
```

7. Piping

```
$ kubectl create configmap greeting \
  --from-file=greeting --dry-run --output json \
  | kubectl apply -f -

configmap "greeting" configured
```

8. Portable Kubeconfig

```
$ kubectl config view

apiVersion: v1
clusters:
- cluster:
    certificate-authority: certificate-authority.crt
    server: https://stage.acme.com:6443
…
```

8. Portable Kubeconfig

```
$ kubectl config view --flatten > portable.kubeconfig

apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJU…
    server: https://stage.acme.com:6443
…
```

9. Current Context

```
$ kubectl config view --minify \
    --output 'jsonpath={..server} {..namespace}'

https://prod.acme.com:6443 mynamespace
```

9. Current Context

```
$ PS1="[\$(
          kubectl config view --minify \
            --output 'jsonpath={..server} {..namespace}'
        )]\$ "

[https://prod.acme.com:6443 mynamespace]$
```

KubeCon

North America 2017

Stupid Kubectl Tricks

Jordan Liggitt, *Red Hat*

http://bit.ly/stupid-kubectl-tricks