Deploying with Docker and Kubernetes

Daniel Fenton

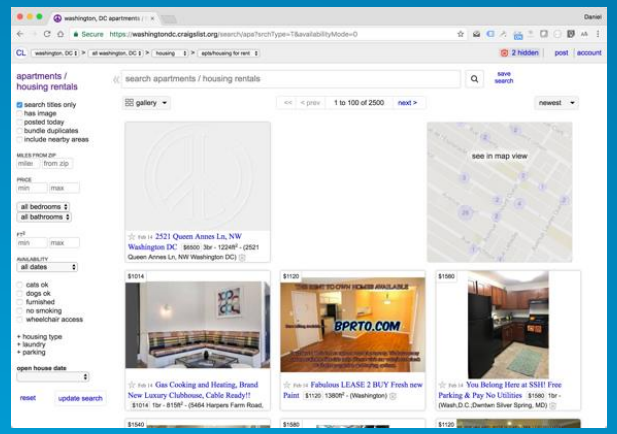
@dmfenton

dfenton@esri.com

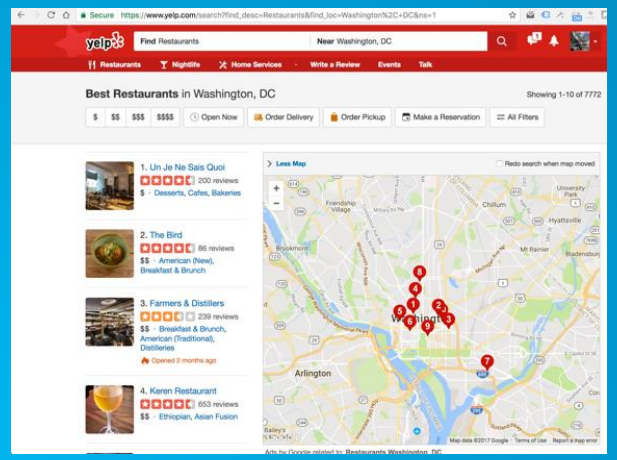
https://github.com/dmfenton

What is Koop?

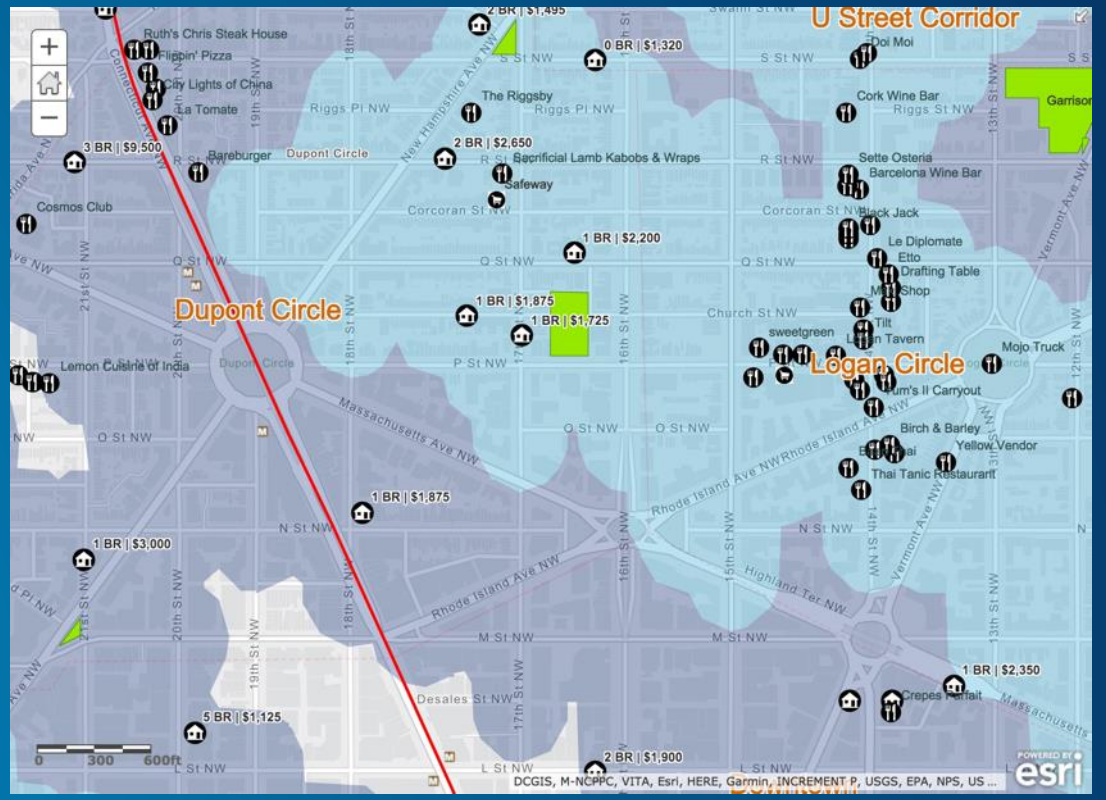
https://washingtondc.craigslist.org/search/apa



https://www.yelp.com/search

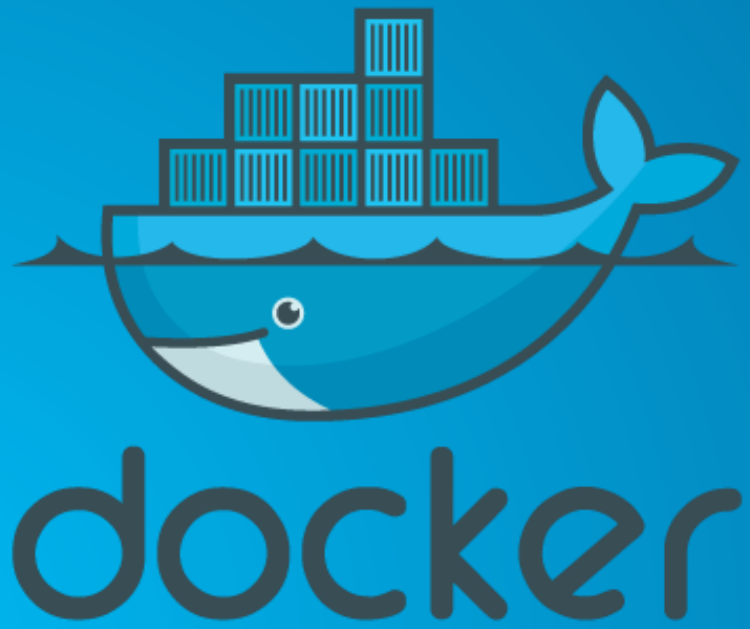


GeoServices



Docker

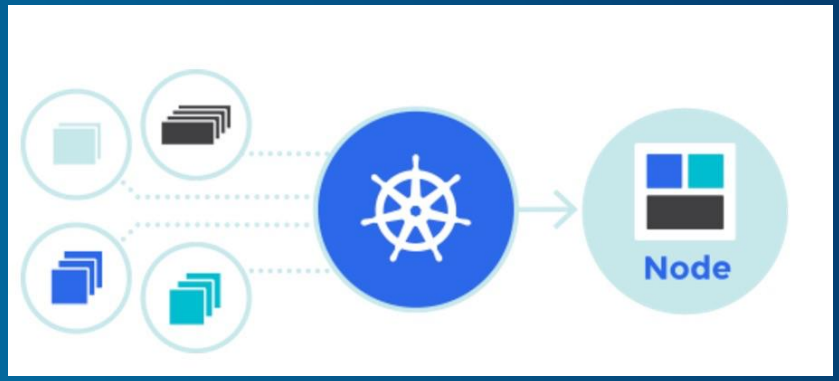
- Package everything you need to run an application into one file
- Meant to solve the “works on my machine” problem
- Lighter than a VM, not a full copy of the OS
- Docker Hub: online container repository




```
FROM node:6
RUN mkdir -p /srv/www/koop
ADD package.json /srv/www/koop
WORKDIR /srv/www/koop
RUN npm install
ADD . /srv/www/koop
EXPOSE 8080
ENTRYPOINT ["/usr/local/bin/node", "/srv/www/koop/server.js"]
```

Kubernetes

- Open Source Container Orchestrator
  - Automate deployment, scaling and operations of containers
- Began life as Google's Borg
- Backed by Google and Microsoft
- Master and nodes architecture
  - Uses etcd under the hood
- Everything specified as yaml files
  - Declare your application and let K8s handle the rest
- https://kubernetes.io/



Kubernetes Cluster Options

- Azure Container Services:
  - Managed service
  - https://docs.microsoft.com/en-us/azure/container-service/container-service-kubernetes-walkthrough
- Google Container Engine
  - Managed service
  - https://cloud.google.com/container-engine/
- AWS
  - Choose your own adventure
    - https://github.com/kubernetes/kops
    - https://kubernetes.io/docs/getting-started-guides/aws/
- Minikube
  - https://kubernetes.io/docs/getting-started-guides/minikube/
  - Local

Kubernetes Components

- Pods: container or containers that form a single logical group
- Replica Set: Supervises the pods
- Deployment: Abstracts replica sets and pods
- Service: Creates an internal and/or external URL and load balances across pods
- Ingress: set of externally available routes that map to services

The Dockerfile

- Runs Node 6 on to of Debian
- Installs and configures Koop
- Saved to docker hub as dmfenton/koop-craigslist
- https://github.com/dmfenton/koop-provider-craigslist/blob/master/Dockerfile

```
FROM node:6
RUN mkdir -p /srv/www/koop
ADD package.json /srv/www/koop
WORKDIR /srv/www/koop
RUN npm install
ADD . /srv/www/koop
EXPOSE 8080
ENTRYPOINT ["/usr/local/bin/node", "/srv/www/koop/server.js"]
```

Kubernetes Deployment

- Which container
- How many replicas
- Readiness check
- Metadata
- https://kubernetes.io/docs/user-guide/deployments/

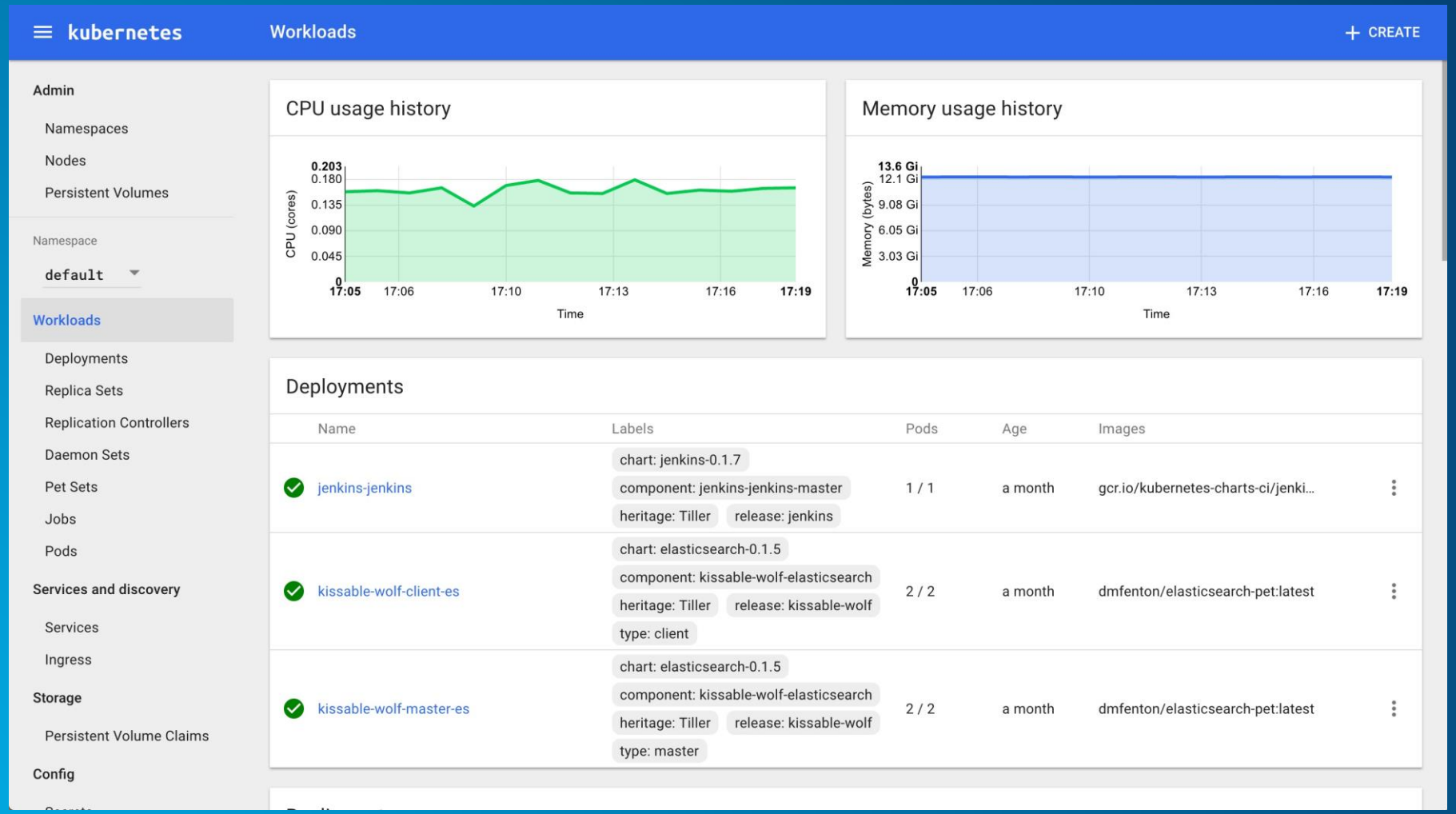
```yaml
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: koop-craigslist
spec:
  replicas: 2
  template:
    metadata:
      labels:
        app: craigslist
    spec:
      containers:
      - name: koop-craigslist
        image: dmfenton/koop-craigslist
        env:
        - name: KOOP_SERVER_PORT
          value: "80"
        ports:
        - containerPort: 80
        livenessProbe:
          httpGet:
            path: /status
            port: 80
          initialDelaySeconds: 10
          timeoutSeconds: 1
        imagePullPolicy: Always
```

Kubernetes Service

- Which port to use
- Installs and configures Koop
- Saved to docker hub as dmfenton/koop-craigslist
- Can automatically assign a load balancer
  - Type => LoadBalancer
- Or, can direct traffic through ingress
  - https://kubernetes.io/docs/user-guide/ingress/

```
28  ---
29  apiVersion: v1
30  kind: Service
31  metadata:
32    name: craigslist
33    labels:
34      app: craigslist
35  spec:
36    type: NodePort
37    ports:
38    - port: 80
39      targetPort: 80
40      protocol: TCP
41      name: http
42    selector:
43      app: craigslist
```

Kubernetes Dashboard



Questions?

Daniel Fenton

@dmfenton

dfenton@esri.com

https://github.com/dmfenton


esri
THE
SCIENCE
OF
WHERE