

Deploying with Docker and Kubernetes

Daniel Fenton

@dmfenton

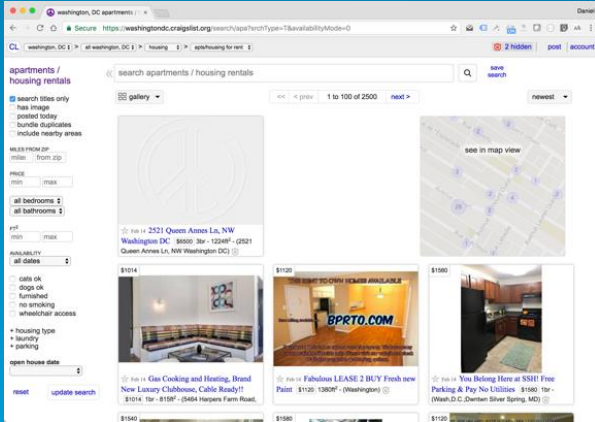
dfenton@esri.com

<https://github.com/dmfenton>

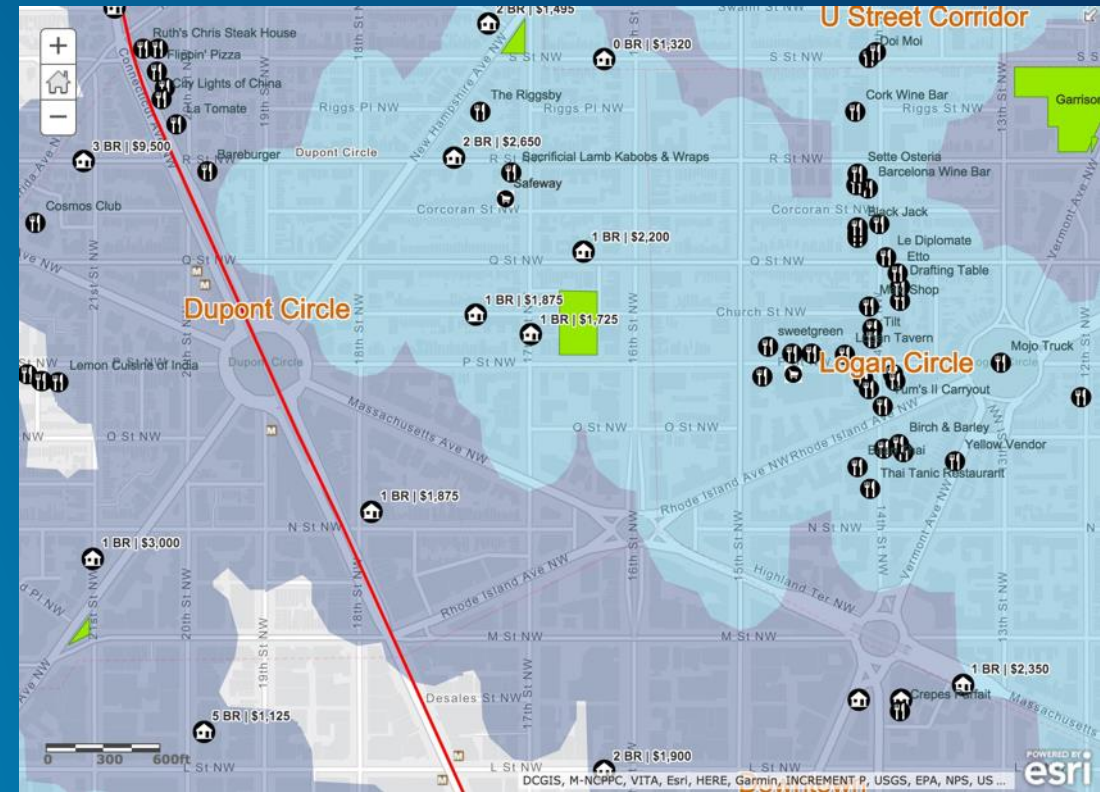


What is Koop?

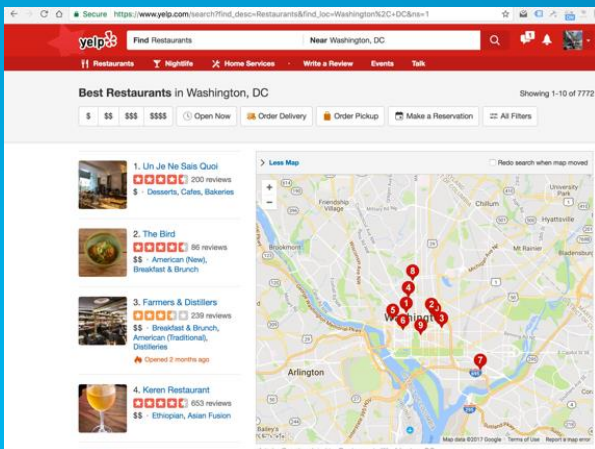
<https://washingtondc.craigslist.org/search/apa>



GeoServices



<https://www.yelp.com/search>



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

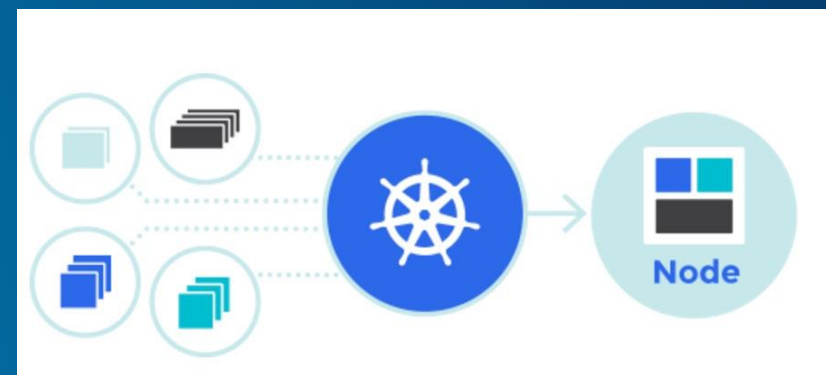


docker

```
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 top 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/>

```
1 ---
2 apiVersion: extensions/v1beta1
3 kind: Deployment
4 metadata:
5   name: koop-craigslist
6 spec:
7   replicas: 2
8   template:
9     metadata:
10    labels:
11      app: craigslist
12    spec:
13      containers:
14      - name: koop-craigslist
15        image: dmfonton/koop-craigslist
16        env:
17        - name: KOOP_SERVER_PORT
18          value: "80"
19        ports:
20        - containerPort: 80
21        livenessProbe:
22          httpGet:
23            path: /status
24            port: 80
25          initialDelaySeconds: 10
26          timeoutSeconds: 1
27        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

☰ **kubernetes** Workloads + CREATE

Admin

- Namespaces
- Nodes
- Persistent Volumes

Namespace

default ▾

Workloads

- Deployments
- Replica Sets
- Replication Controllers
- Daemon Sets
- Pet Sets
- Jobs
- Pods

Services and discovery

- Services
- Ingress

Storage

- Persistent Volume Claims

Config

CPU usage history

Time	CPU (cores)
17:05	0.135
17:06	0.140
17:07	0.145
17:08	0.135
17:09	0.140
17:10	0.150
17:11	0.160
17:12	0.140
17:13	0.150
17:14	0.140
17:15	0.145
17:16	0.140
17:17	0.145
17:18	0.140
17:19	0.145

Memory usage history

Time	Memory (bytes)
17:05	12.1 Gi
17:06	12.1 Gi
17:07	12.1 Gi
17:08	12.1 Gi
17:09	12.1 Gi
17:10	12.1 Gi
17:11	12.1 Gi
17:12	12.1 Gi
17:13	12.1 Gi
17:14	12.1 Gi
17:15	12.1 Gi
17:16	12.1 Gi
17:17	12.1 Gi
17:18	12.1 Gi
17:19	12.1 Gi

Deployments

Name	Labels	Pods	Age	Images	
✓ jenkins-jenkins	chart: jenkins-0.1.7 component: jenkins-jenkins-master heritage: Tiller release: jenkins	1 / 1	a month	gcr.io/kubernetes-charts-ci/jenki...	⋮
✓ kissable-wolf-client-es	chart: elasticsearch-0.1.5 component: kissable-wolf-elasticsearch heritage: Tiller release: kissable-wolf type: client	2 / 2	a month	dmfenton/elasticsearch-pet:latest	⋮
✓ kissable-wolf-master-es	chart: elasticsearch-0.1.5 component: kissable-wolf-elasticsearch heritage: Tiller release: kissable-wolf type: master	2 / 2	a month	dmfenton/elasticsearch-pet:latest	⋮

Questions?

Daniel Fenton

@dmfenton

dfenton@esri.com

<https://github.com/dmfenton>



esri

THE
SCIENCE
OF
WHERE