Deploy Like a Boss: Using Apache® Ignite™ and Kubernetes®

Dani Traphagen
Solutions Architect
GridGain

http://ignite.apache.org

#apacheignite

Agenda

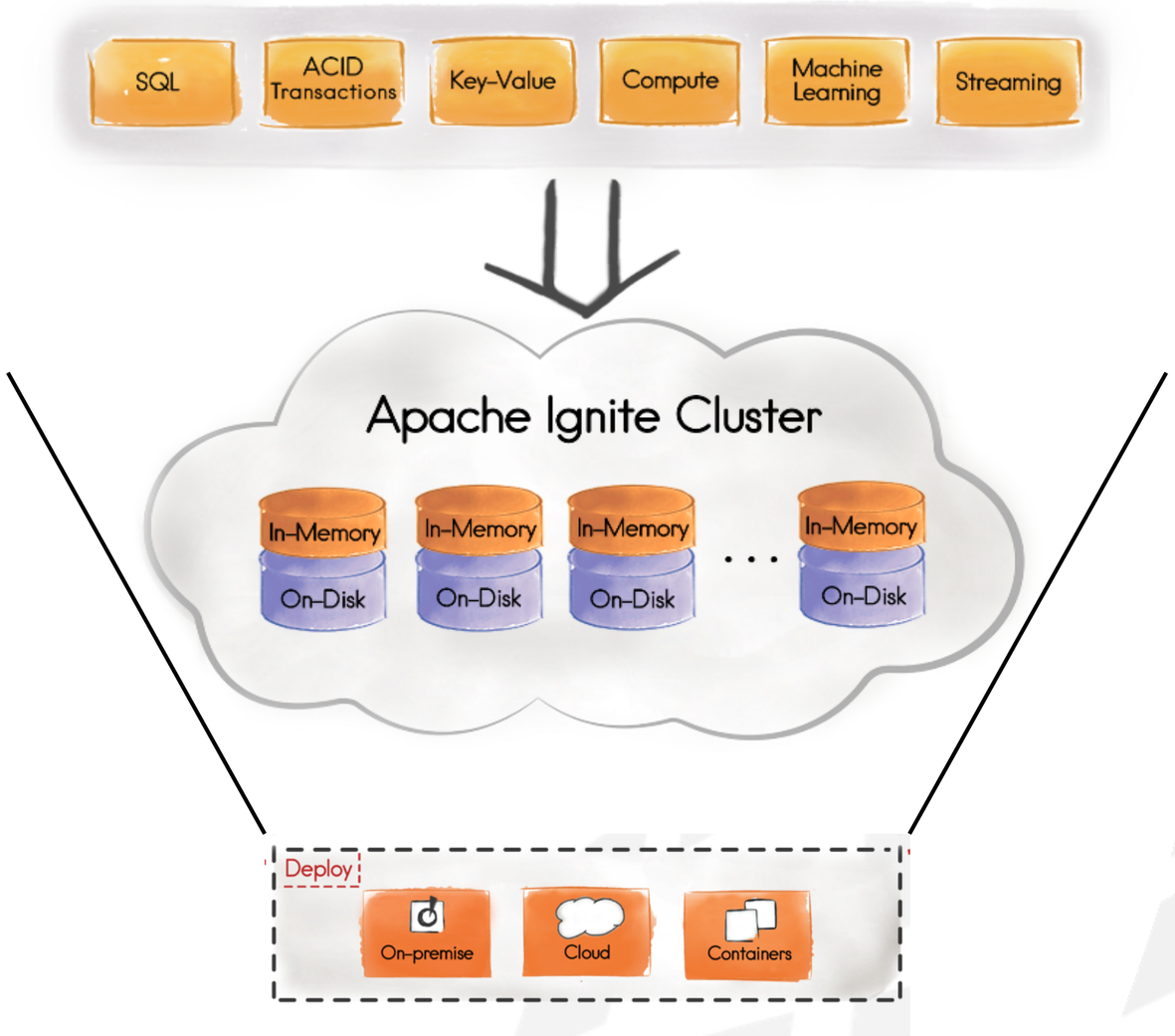
- Setting up a Apache Ignite cluster
- Using the Kubernetes IP Finder and the Kubernetes Ignite Lookup Service
- Sharing the Ignite Cluster Configuration
- Deploying your Ignite Pods
- Adjusting the Ignite Cluster Size when you need to Scale
  - Try it out!

By the end – this will be you..

First - Apache Ignite's Platform

Apache Ignite Platform (Version 2.1 Depicted)

Kubernetes works with versions 1.9 and above due to

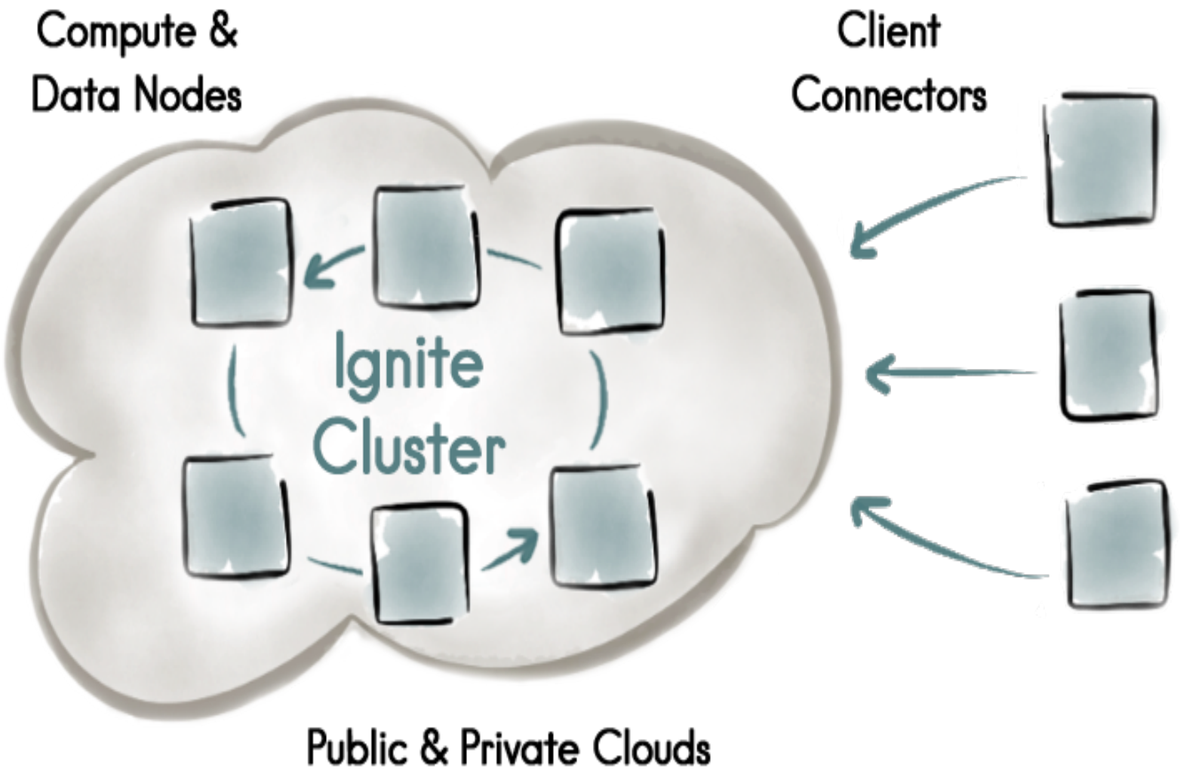


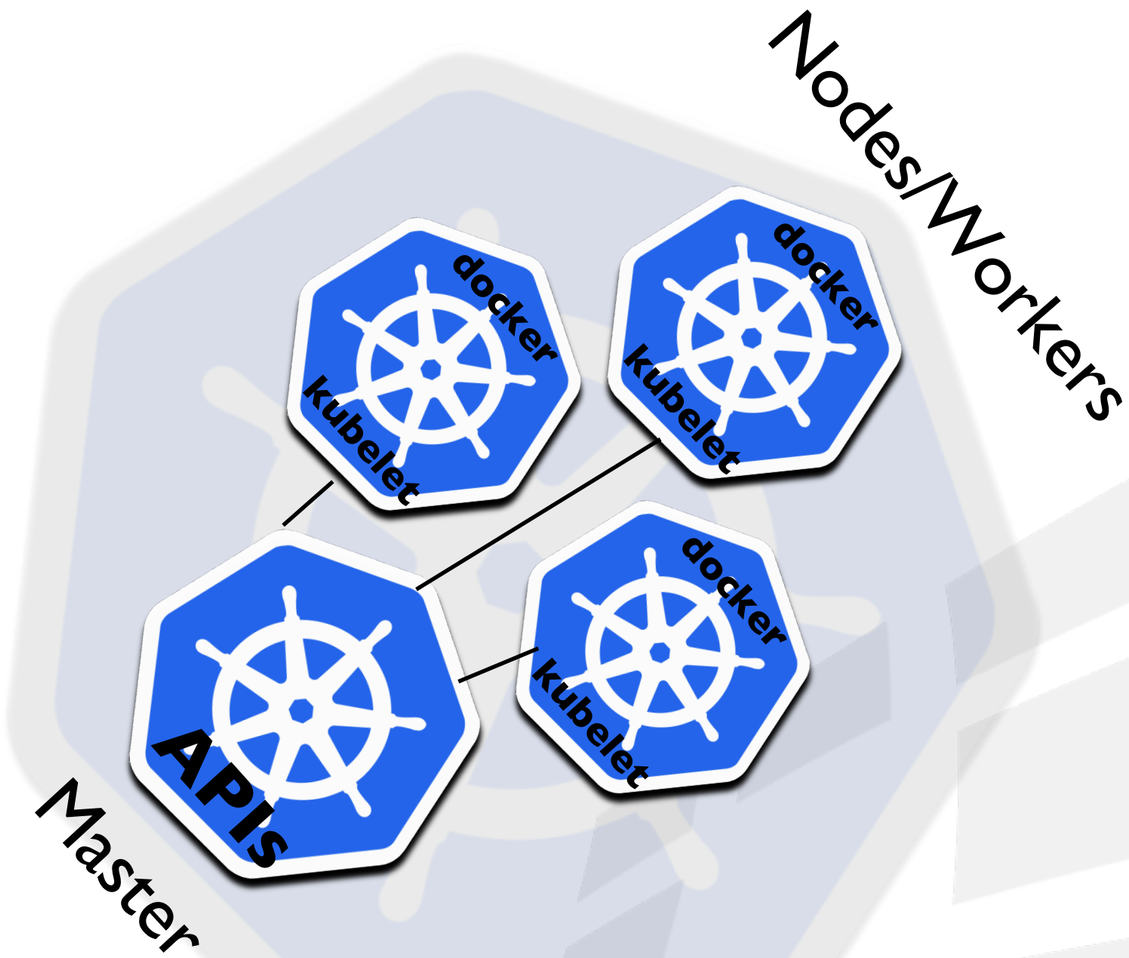
Second – Deployment with Kubernetes (K8)

Kubernetes in a Simple Definition

- “Kubernetes intends to radically simplify the task of building, deploying and maintaining distributed systems.”
  - Kubernetes: Up and Running: Dive into the Future of Infrastructure
    - By: Kelsey Hightower

Apache Ignite + K8 Cluster Architecture





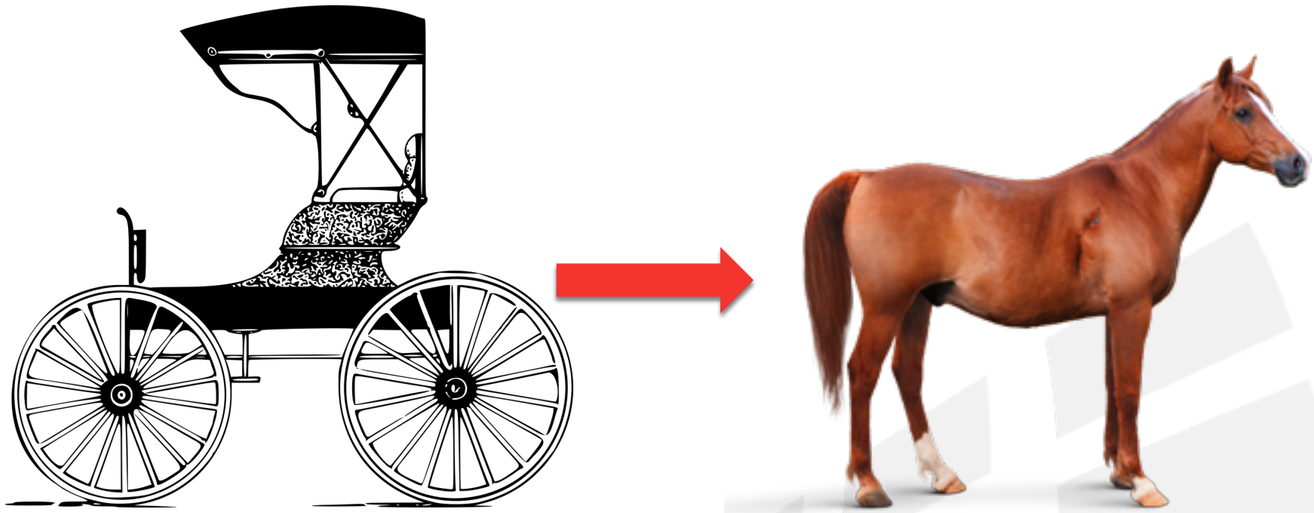
Benefits of K8 (version 1.7)

- Cost Efficiency
  - Use of containers by multiple developers ensures shared resourcing rather than redundancy
- High Availability and Performance
  - Deploying with K8 and setting rules for a set number of nodes in AWS will ensure your cluster can always handle the transactions hitting it
  - No more pager duty

Setting Up an Apache Ignite Cluster

Step 1:

- What is your use case?
- Seriously, what is your use case?
- I'm not kidding. Use case, then set up…
  - Cart before horse please.



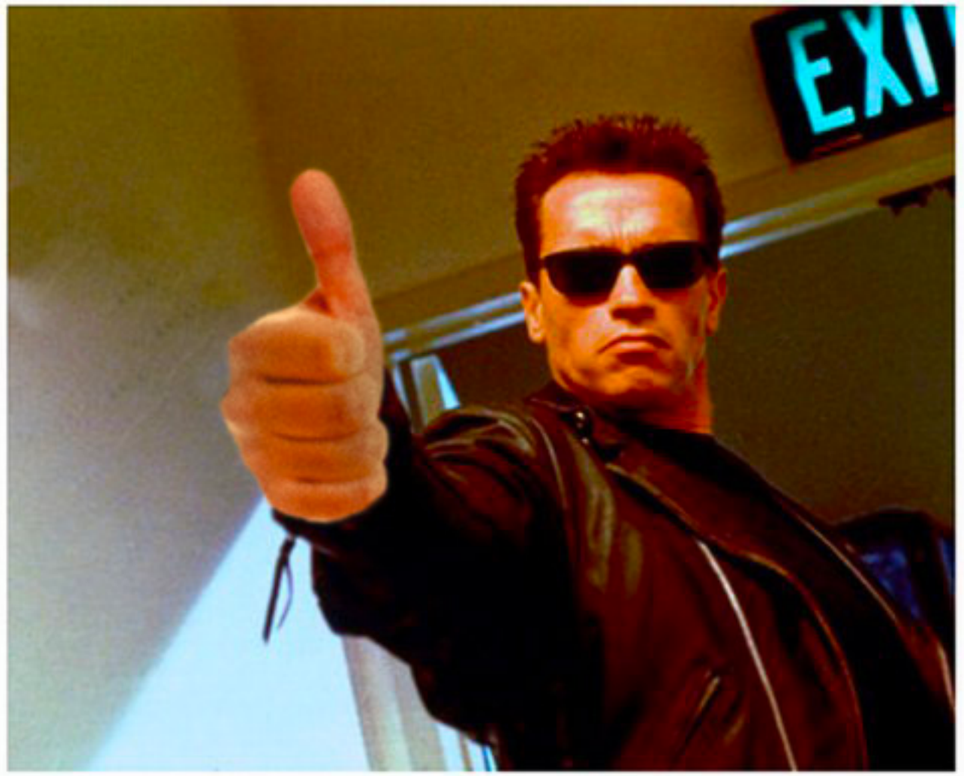
Setting Up an Apache Ignite Cluster

- 1. Download Apache Ignite
- 2. Make sure to add the ignite-kubernetes Maven Dependency to your pom.xml
- 3. I'm using examples from our docs.

```
<!--
    Ignite configuration with all defaults and enabled p2p deployment and enabled events.
-->
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="
        http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd
        ">


    <bean class="org.apache.ignite.configuration.IgniteConfiguration">
        <!-- Enabling the peer-class loading feature. -->
        <property name="peerClassLoadingEnabled" value="true"/>

        <!--
          Labeling Data Nodes with special attribute. This attribute is checked by common.filters.DataNodeFilters
          which decides where caches have to be deployed.
```

Setting up K8 on a Local Machine

- Install K8 where you intend
  - AWS
  - Google Cloud
  - Dev Machine
- Set your $PATH w/K8
  - Install Kubectl
  - I used Brew



Kubernetes Discovery

- Multicast = ☹
  - Use Static IP Finder & list Ignite IPs, K8 will dynamically assign them
- You can use other cloud Ignite IP finders but you need K8 to running in the cloud env.
- What's the point of the `TcpDiscoveryKubernetesIpFinder`?

Using the Kubernetes IP Finder and the Kubernetes Ignite Lookup Service

```
<property name="discoverySpi">
    <bean class="org.apache.ignite.spi.discovery.tcp.TcpDiscoverySpi">
        <property name="ipFinder">
            <bean class="org.apache.ignite.spi.discovery.tcp.ipfinder.kubernetes.TcpDiscoveryKubernetesIpFinder">
            </bean>
        </property>
    </bean>
```

- Apps & nodes running outside of K8 & Ignite will not be able to reach the cluster
- K8 service should be deployed before Ignite cluster boot
- The Ignite Pods internal IPs will be maintained by the K8 service.
  - Service name must be equal to `setServiceName(String)`
  - This will be `ignite` as a default

Using Docker on Local Machine with Minikube

```
danitraphagen@Dani-MBP ~$ docker ps

CONTAINER ID        IMAGE                                                COMMAND                  CREATED             STATUS
 PORTS               NAMES
97cc92b952b8        gcr.io/google_containers/kubernetes-dashboard-amd64    "/dashboard --inse..."   3 minutes ago       Up 3 minutes
                     k8s_kubernetes-dashboard_kubernetes-dashboard-xc9h2_kube-system_3e340ea3-87b9-11e7-9121-080027e5d2cc_0
abeb3cac70c2        gcr.io/google_containers/pause-amd64:3.0             "/pause"                 4 minutes ago       Up 4 minutes
                     k8s_POD_kubernetes-dashboard-xc9h2_kube-system_3e340ea3-87b9-11e7-9121-080027e5d2cc_0
6a0a5bd1fe67        gcr.io/google_containers/k8s-dns-dnsmasq-nanny-amd64   "/dnsmasq-nanny -v..."   35 minutes ago      Up 35 minutes
                     k8s_dnsmasq_kube-dns-910330662-98hwb_kube-system_cc653490-87b4-11e7-9121-080027e5d2cc_0
61c0b0ec07d4        gcr.io/google_containers/k8s-dns-kube-dns-amd64       "/kube-dns --domai..."   35 minutes ago      Up 35 minutes
                     k8s_kubedns_kube-dns-910330662-98hwb_kube-system_cc653490-87b4-11e7-9121-080027e5d2cc_0
02282e7c7c22        gcr.io/google_containers/k8s-dns-sidecar-amd64        "/sidecar --v=2 --..."   35 minutes ago      Up 35 minutes
                     k8s_sidecar_kube-dns-910330662-98hwb_kube-system_cc653490-87b4-11e7-9121-080027e5d2cc_0
c4d5a18dcd4d        gcr.io/google_containers/pause-amd64:3.0             "/pause"                 35 minutes ago      Up 35 minutes
                     k8s_POD_kube-dns-910330662-98hwb_kube-system_cc653490-87b4-11e7-9121-080027e5d2cc_0
b6743ee53b24        gcr.io/google_containers/echoserver                  "nginx -g 'daemon ..."   About an hour ago   Up 47 minutes
                     k8s_hello-minikube_hello-minikube-180744149-9n0g5_default_36b044eb-87b3-11e7-9121-080027e5d2cc_0
d8d59a7c5325        gcr.io/google_containers/pause-amd64:3.0             "/pause"                 About an hour ago   Up 47 minutes
                     k8s_POD_hello-minikube-180744149-9n0g5_default_36b044eb-87b3-11e7-9121-080027e5d2cc_0
fa27860f8345        gcr.io/google-containers/kube-addon-manager          "/opt/kube-addons.sh"    About an hour ago   Up 55 minutes
                     k8s_kube-addon-manager_kube-addon-manager-minikube_kube-system_c654b2f084cf26941c334a2c3d6db53d_0
157a056988b7        gcr.io/google_containers/pause-amd64:3.0             "/pause"                 About an hour ago   Up 55 minutes
                     k8s_POD_kube-addon-manager-minikube_kube-system_c654b2f084cf26941c334a2c3d6db53d_0
danitraphagen@Dani-MBP ~$
```

Sharing the Ignite Cluster Configuration

Service Startup & Sharing Configs

- Minikube - Two basic commands
  - minikube start
  - minikube dashboard
- Kubectl
  - kubectl create -f ~/<path-to-project>/<project-name>/config/ignite-service.yaml

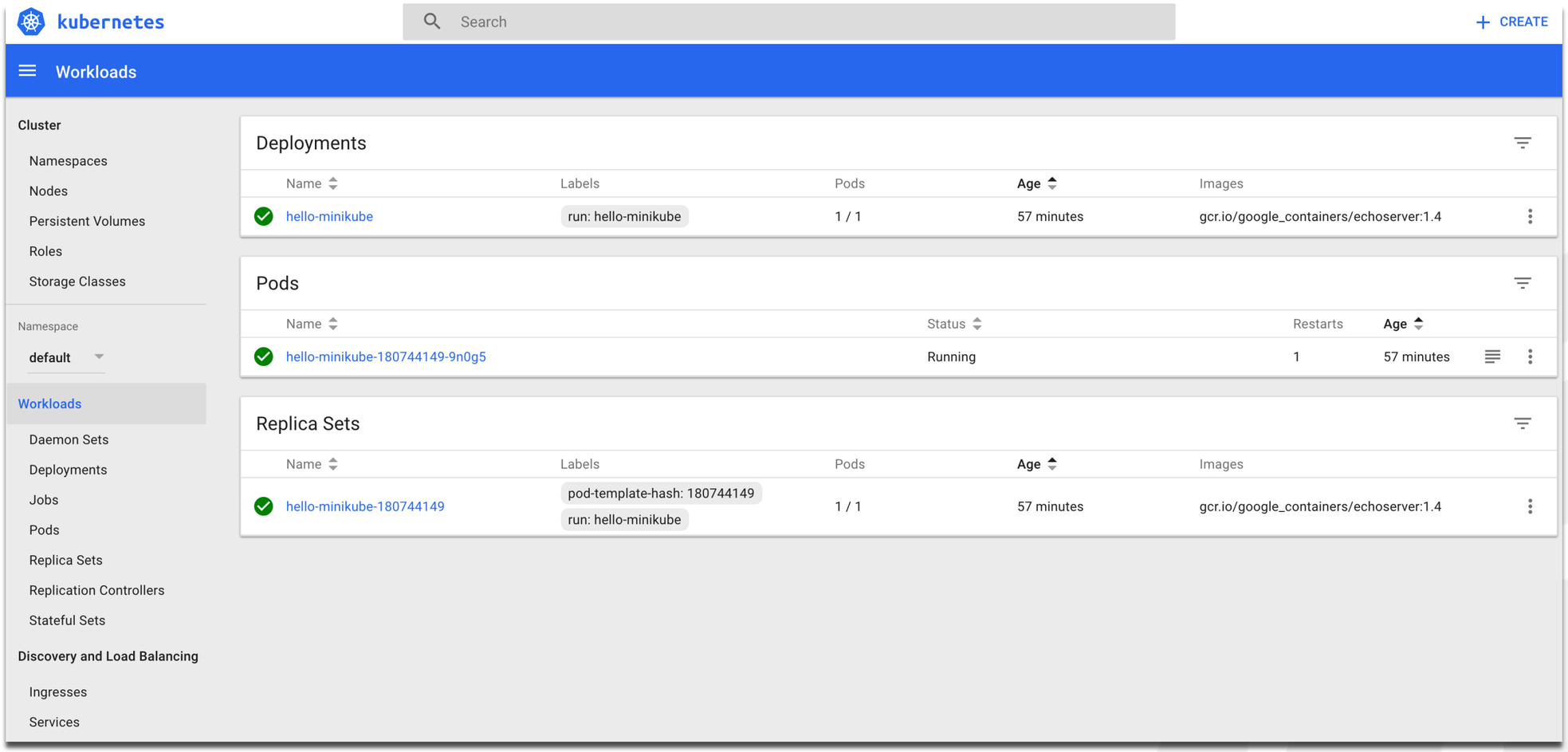
```
service "ignite" created
danitraphagen@Dani-MBP ~$
```

```
danitraphagen@Dani-MBP ~$ minikube start
Starting local Kubernetes v1.7.0 cluster...
Starting VM...
Getting VM IP address...
Moving files into cluster...
Setting up certs...
Starting cluster components...
Connecting to cluster...
Setting up kubeconfig...
Kubectl is now configured to use the cluster.
danitraphagen@Dani-MBP ~$ minikube dashboard
Opening kubernetes dashboard in default browser...
danitraphagen@Dani-MBP ~$
```

Yes, someone I follow on Twitter did this.

Now Kubernetes is Running!

- Dashboard is in business!



Configuring your Ignite Pods

- **2 Things Needed!**
  - 1. Apache Ignite Configuration File with the Kubernetes IP Finder
  - 2. YAML Configurations for the Apache Ignite pods/ nodes
- **Steps:**
- **Create your ignite-service.yaml**
  - kubectl create -f ignite-service.yaml
  - kubectl get svc ignite

```
NAME      CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
ignite    None         <none>        9042/TCP   29s
```

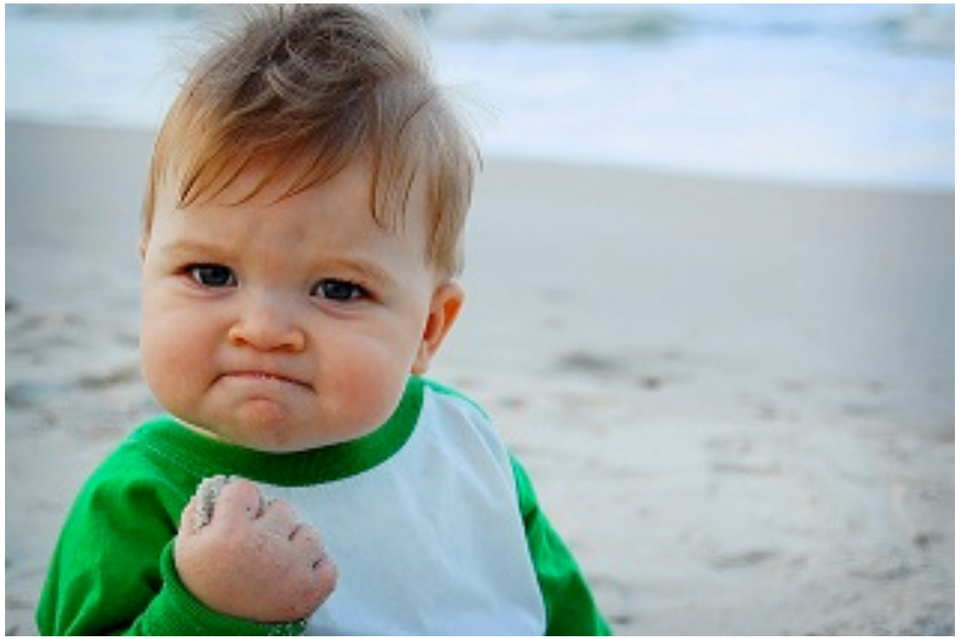
Sharing Ignite Cluster Configs

- Confirm the ignite service was created
- Make a path to the persistence volume docker will use to pass the kubernetes config ‘example-kube.xml’

```
danitraphagen@Dani-MBP ~$  kubectl get svc ignite

NAME      CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
ignite    None         <none>        9042/TCP   13m
```

```
danitraphagen@Dani-MBP ~/data/ignite$ kubectl create -f ignite-volume.yaml

persistentvolume "ignite-volume" created
```

Success



```
danitraphagen@Dani-MBP ~/data/ignite$ kubectl get pv ignite-volume

NAME             CAPACITY   ACCESSMODES   RECLAIMPOLICY   STATUS
ignite-volume    1Gi        RWO           Retain          Available
danitraphagen@Dani-MBP ~/data/ignite$
```

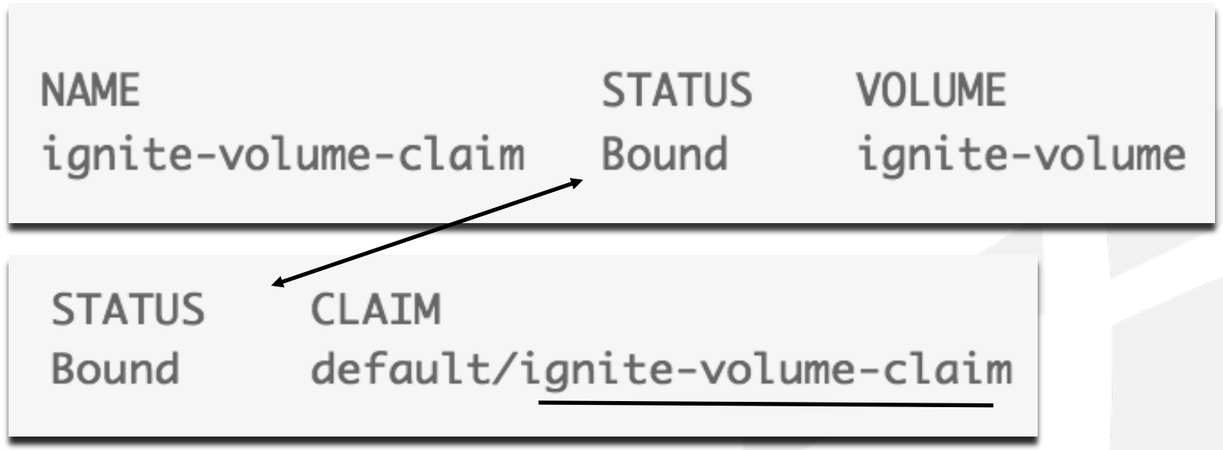
Now, do this for the ignite-volume-claim.yaml

ignite-volume-claim.yaml

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: ignite-volume-claim
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

Persistent Volume Bound?

- Make sure your persistent volume is bound to the claim
  - kubectl get pvc ignite-volume-claim
  - kubectl get pv ignite volume



Deploying your Ignite Pods

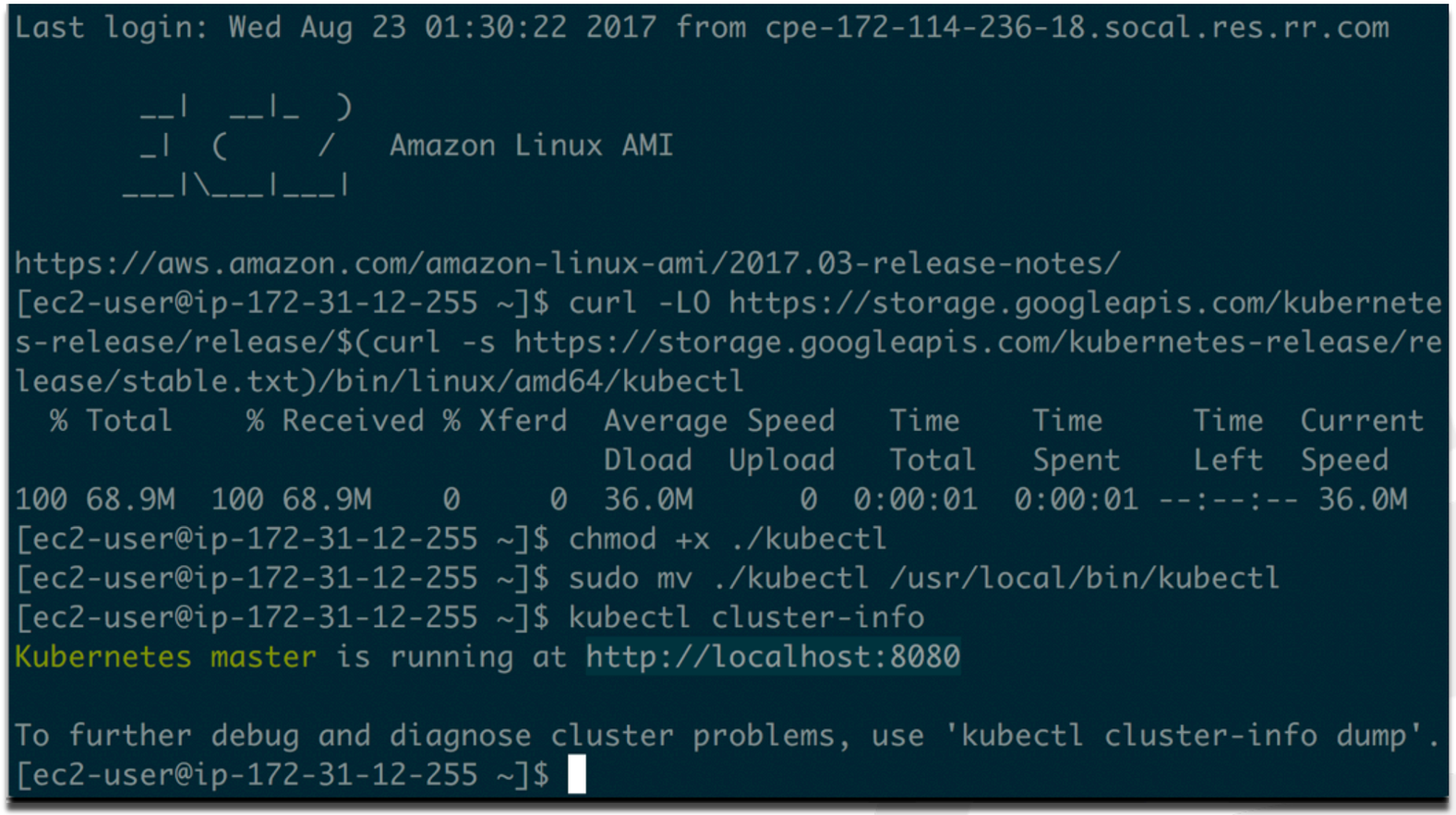
- Now it’s time to launch
  - kubectl create -f ignite-deployment.yaml
  - kubectl get pods
- Get the logs and examine for each cluster…ex)
  - kubectl logs ignite-cluster-3454482164-d4m6g
- Scale out:
  - kubectl scale --replicas=5 -f ignite-deployment.yaml

```
NAME                                READY     STATUS    RESTARTS   AGE
ignite-cluster-3454482164-d4m6g     1/1       Running   0          25m
ignite-cluster-3454482164-w0xtx     1/1       Running   0          25m
```

```
NAME                                READY     STATUS    RESTARTS   AGE
ignite-cluster-3454482164-d4m6g     1/1       Running   0          34m
ignite-cluster-3454482164-ktkrr     1/1       Running   0          58s
ignite-cluster-3454482164-r20f8     1/1       Running   0          58s
ignite-cluster-3454482164-vf8kh     1/1       Running   0          58s
ignite-cluster-3454482164-w0xtx     1/1       Running   0          34m
```

Many Deployment Options…

- On Premise
- Cloud
  - Azure
  - EC2
  - Google Cloud



Using Minikube for Local Dev

- A good place to start for exploration
- When you want to get in the cloud - pick your poison
  - Post on setup w/Azure by Ignite PMC Denis Magda:
    - https://dzone.com/articles/deploying-apache-ignite-in-kubernetes-on-microsoft
    - https://kubernetes.io/docs/setup/pick-right-solution/#turnkey-cloud-solutions

```
danitraphagen@Dani-MBP ~$ curl -Lo minikube https://storage.googleapis.com/minik
ube/releases/v0.21.0/minikube-darwin-amd64 && chmod +x minikube && sudo mv minik
ube /usr/local/bin/

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 81.9M  100 81.9M    0     0  8189k      0  0:00:10  0:00:10 --:--:-- 8865k
danitraphagen@Dani-MBP ~$ minikube start
Starting local Kubernetes v1.7.0 cluster...
Starting VM...
Downloading Minikube ISO
 97.80 MB / 97.80 MB [============================================] 100.00% 0s
Getting VM IP address...
Moving files into cluster...
Setting up certs...
Starting cluster components...
Connecting to cluster...
Setting up kubeconfig...
Kubectl is now configured to use the cluster.
```

Dare to try?

1. Deploy Cloud Environment

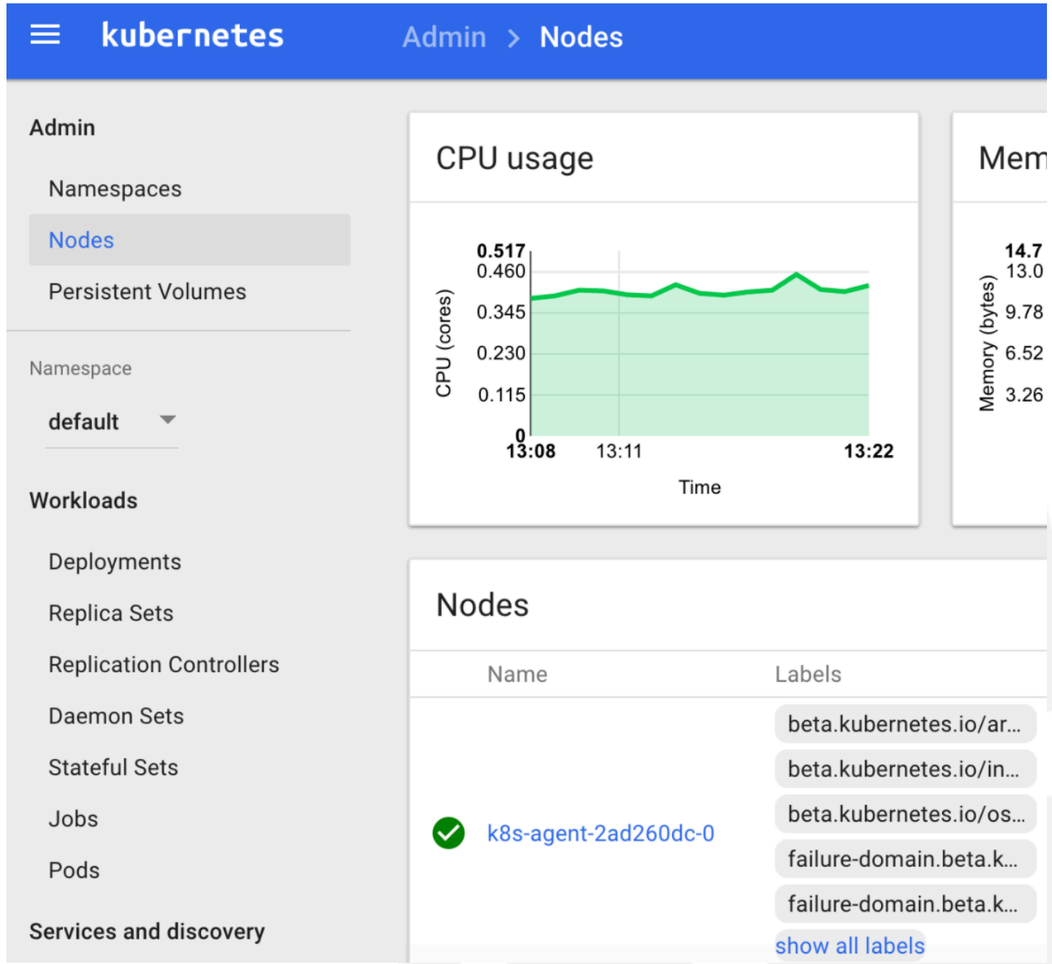
```
dani@Azure:~$ kubectl get nodes
NAME                   STATUS                        AGE       VERSION
k8s-agent-747b6f74-0   NotReady                      15s       v1.6.6
k8s-agent-747b6f74-1   NotReady                      15s       v1.6.6
k8s-agent-747b6f74-2   NotReady                      19s       v1.6.6
k8s-master-747b6f74-0  NotReady,SchedulingDisabled   17s       v1.6.6
dani@Azure:~$ kubectl get nodes
NAME                   STATUS                        AGE       VERSION
k8s-agent-747b6f74-0   Ready                         1m        v1.6.6
k8s-agent-747b6f74-1   Ready                         1m        v1.6.6
k8s-agent-747b6f74-2   Ready                         1m        v1.6.6
k8s-master-747b6f74-0  Ready,SchedulingDisabled      1m        v1.6.6
dani@Azure:~$
```

2. Connect to your Cloud Environment from your local machine, ex) Azure

```
#connect to cluster, make sure your sshkeys are setup

az acs kubernetes get-credentials --resource-group=myResourceGroup --name=myK8sCluster

#make sure you see the k8s-agents & master

kubectl get nodes
```

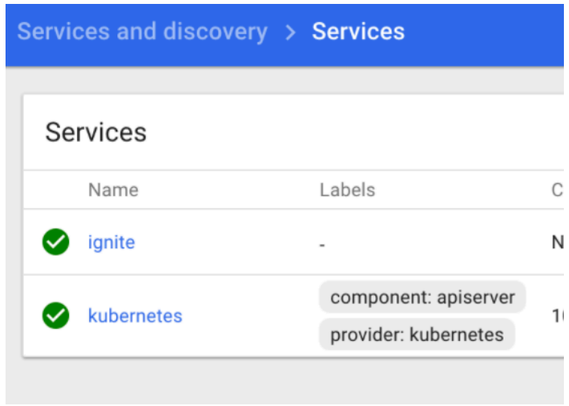
3. Using the Dashboard

1. kubectl proxy
2. http://localhost:8001/ui



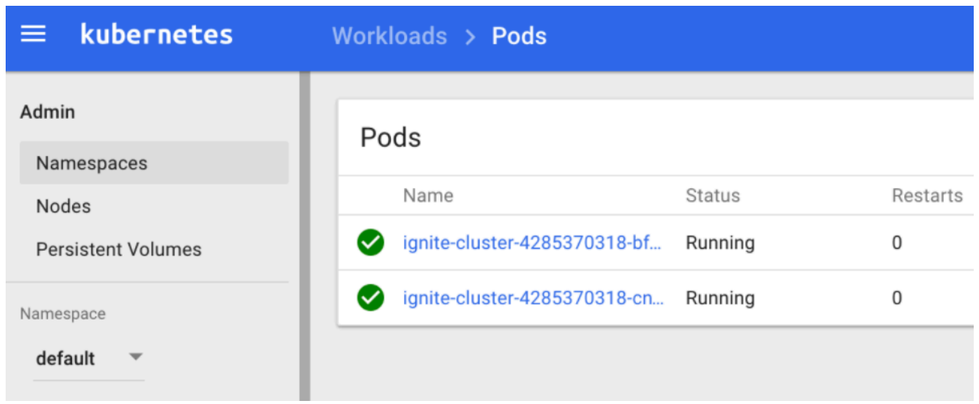
4. Create the K8 Lookup Service

```
#using above link, create the file then
initiate the service

kubectl create -f ignite-service.yaml

#you will see that the ignite service is under
the services tab
```



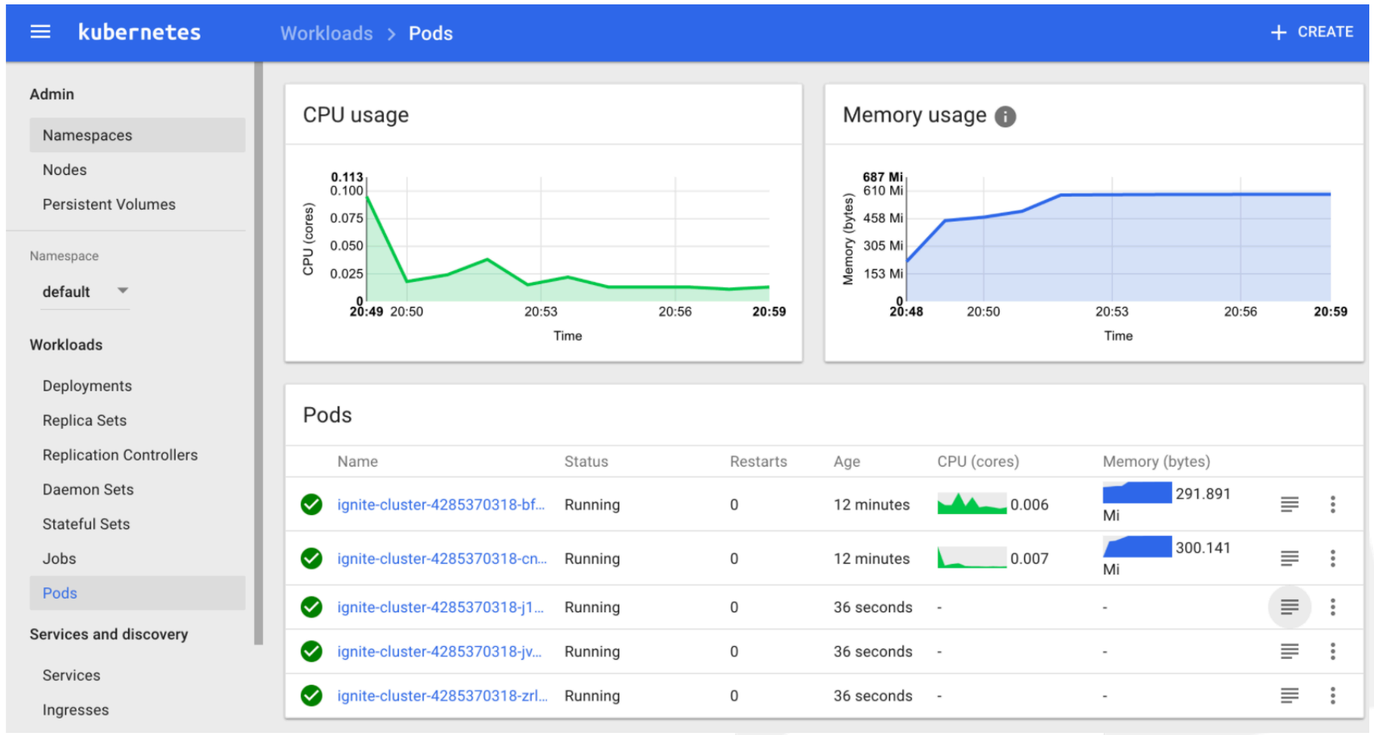
5. Deploy your Apache Ignite Cluster

```
#create the ignite-deployment.yaml file following instructions here

kubectl create -f ignite-deployment.yaml

#you will see that the ignite cluster is running in kubernetes
```



Adjusting the Ignite Cluster Size when you Need to Scale

- When you want to elasticly scale out your cluster with K8:
  - `kubectl scale –replicas=<n> -f ignite-deployment.yaml`
- run
  - `kubectl get pods`
- Let's say you want 5 nodes?
  - `kubectl scale --replicas=5 -f ~/kubernetes_dev/azure/ignite-deployment.yaml`
- You will see your cluster scale out – in this case from 2 to 5 nodes!



Overall Steps

- Ignite Download:
  - https://ignite.apache.org/download.cgi
- Run Kubernetes Locally:
  - https://kubernetes.io/docs/getting-started-guides/minikube/
- Deploy Kubernetes & Ignite
  - https://apacheignite.readme.io/docs/kubernetes-deployment

Resources

- Denis Magda's post on deploying Ignite/K8 in Azure:
  - https://dzone.com/articles/deploying-apache-ignite-in-kubernetes-on-microsoft
- Tutorials
  - https://kubernetes.io/docs/tutorials/
- K8 Book by Kelsey Hightower:
  - http://shop.oreilly.com/product/0636920043874.do
- Ignite Book by Shahim and others:
  - https://leanpub.com/ignite

ANY QUESTIONS?

Thank you for joining us. Follow the conversation.

http://ignite.apache.org

#apacheignite