

P.PORTO

Methods and Techniques for
Software Development

Ricardo Santos | 2019/2020
rjs@estg.ipp.pt

P.PORTO

Kubernetes Aula 5

Methods and Techniques for
Software Development

2019/2020

P.PORTO

Kubernetes

Lesson 5

Methods and Techniques for
Software Development

2019/2020



kubernetes

Hands on lab

Kubernetes

- Exercise - ActiveMQ
- ReplicaSets
- Deployments

Exercise

Deploy image:

`richardchesterwood/k8s-fleetman-queue (release1)`

Port **8161** is the admin console
(admin/admin)

Expose this to a browser using **30010**

▶ kubectl get all

```
$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/webapp	1/1	Running	2	3d
pod/webapp-release-0-5	1/1	Running	1	2d

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP	2d
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d

▶ Create a new file called pods.yaml

► pods.yaml

```
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: webapp
5    labels:
6      app: webapp
7      release: "0"
8
9  spec:
10   containers:
11     - name: webapp
12       image: richardchesterwood/k8s-fleetman-webapp-angular:release0
13
14   ---
15
```

```
16    apiVersion: v1
17    kind: Pod
18    metadata:
19      name: webapp-release-0-5
20      labels:
21        app: webapp
22        release: "0-5"
23
24    spec:
25      containers:
26      - name: webapp
27        image: richardchesterwood/k8s-fleetman-webapp-angular:release0-5
28
```

```
29    ---  
30    apiVersion: v1  
31    kind: Pod  
32    metadata:  
33    |   name: queue ←  
34    |   labels:  
35    |   |   app: queue  
36    spec:  
37    |   containers:  
38    |   |   - name: queue  
39    |   |   |   image: richardchesterwood/k8s-fleetman-queue:release1  
40
```

- ▶ List your directory - `ls`

```
$ ls
pods.yaml  webapp-service.yaml
```

- ▶ Rename your `webapp-service.yaml` to `services.yaml`

► services.yaml

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: fleetman-webapp
5
6  spec:
7    selector:
8      app: webapp
9      release: "0-5"
10
11  ports:
12    - name: http
13      port: 80
14      nodePort: 30080
15
16  type: NodePort
17
```

```
18    ---
19    apiVersion: v1
20    kind: Service
21    metadata:
22      name: fleetman-queue
23
24    spec:
25      selector:
26        app: queue
27
28      ports:
29        - name: http
30          port: 8161 ←
31          nodePort: 30010 ←
32
33      type: NodePort
34
```

▶ **kubectl apply -f .**

```
$ kubectl apply -f .
pod "webapp" unchanged
pod "webapp-release-0-5" unchanged
pod "queue" unchanged
service "fleetman-webapp" unchanged
service "fleetman-queue" created
```

▶ **kubectl get all**

NAME					
	READY	STATUS	RESTARTS	AGE	
pod/queue	1/1	Running	0	4m	←
pod/webapp	1/1	Running	2	3d	
pod/webapp-release-0-5	1/1	Running	1	3d	

NAME					
	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP	55s
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP	3d
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d

► kubectl describe pod queue

Events:				
Type	Reason	Age	From	Message
Normal	Scheduled	4m	default-scheduler	Successfully assigned queue to minikube
Normal	SuccessfulMountVolume	4m	kubelet, minikube	MountVolume.SetUp succeeded for volume "default-token-wpbqh"
Normal	Pulling	4m	kubelet, minikube	pulling image "richardchesterwood/k8s-fleetman-queue:release1"
Normal	Pulled	3m	kubelet, minikube	Successfully pulled image "richardchesterwood/k8s-fleetman-queue:release1"
Normal	Created	3m	kubelet, minikube	Created container
Normal	Started	3m	kubelet, minikube	Started container

Let's try it now!

The screenshot shows a web browser window with the URL `192.168.99.100:30010` in the address bar. The main content is the Apache ActiveMQ welcome page. At the top, there is a logo of a white bird in flight on a dark background. Below the logo, the word "Active" is in purple and "MQ" is in black, with a large shadow effect underneath. A horizontal purple bar runs across the middle of the page. Below this bar, the text "Welcome to the Apache ActiveMQ!" is displayed in bold black font. Underneath, a question "What do you want to do next?" is followed by two bullet points: "■ Manage ActiveMQ broker" and "■ See some Web demos (demos not included in default configuration)". At the bottom of the page, a black footer bar contains the text "Copyright 2005-2015 The Apache Software Foundation." and "Graphic Design By Hiram".

← → C ⓘ 192.168.99.100:30010

ActiveMQ

Welcome to the Apache ActiveMQ!

What do you want to do next?

- Manage ActiveMQ broker
- See some Web demos (demos not included in default configuration)

Copyright 2005-2015 The Apache Software Foundation.

Graphic Design By Hiram



kubernetes

ReplicaSet

Pods are very basic and disposable objects!

- ▶ In the future - in a production system - is more likely for you to work with
 - ▶ Deployments
 - ▶ ReplicaSets

- ▶ In a full running system is more than possible that pods are going to die
 - ▶ If a node fails - pod dies
 - ▶ If a pod consumes too many resources - kubernetes kills the pod

Pods can be short-lived

- ▶ If you deploy a pod directly like we've been doing - ***you are responsible for the lifetime of that pod as well as the welfare of that pod***

NAME	READY	STATUS	RESTARTS	AGE
pod/queue	1/1	Running	0	33m
pod/webapp	1/1	Running	2	3d
pod/webapp-release-0-5	1/1	Running	1	3d

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP	30m
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP	3d
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d

- ▶ If any of this pods die, they are not going to come back!

Let's demonstrate that!

▶ kubectl get all

```
$ kubectl get all
NAME                         READY   STATUS    RESTARTS   AGE
pod/queue                     1/1     Running   0          33m
pod/webapp                    1/1     Running   2          3d
pod/webapp-release-0-5        1/1     Running   1          3d

NAME                           TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)           AGE
service/fleetman-queue         NodePort    10.100.30.30   <none>       8161:30010/TCP   30m
service/fleetman-webapp       NodePort    10.107.11.35   <none>       80:30080/TCP    3d
service/kubernetes            ClusterIP   10.96.0.1      <none>       443/TCP          4d
```



▶ kubectl svc fleetman-webapp

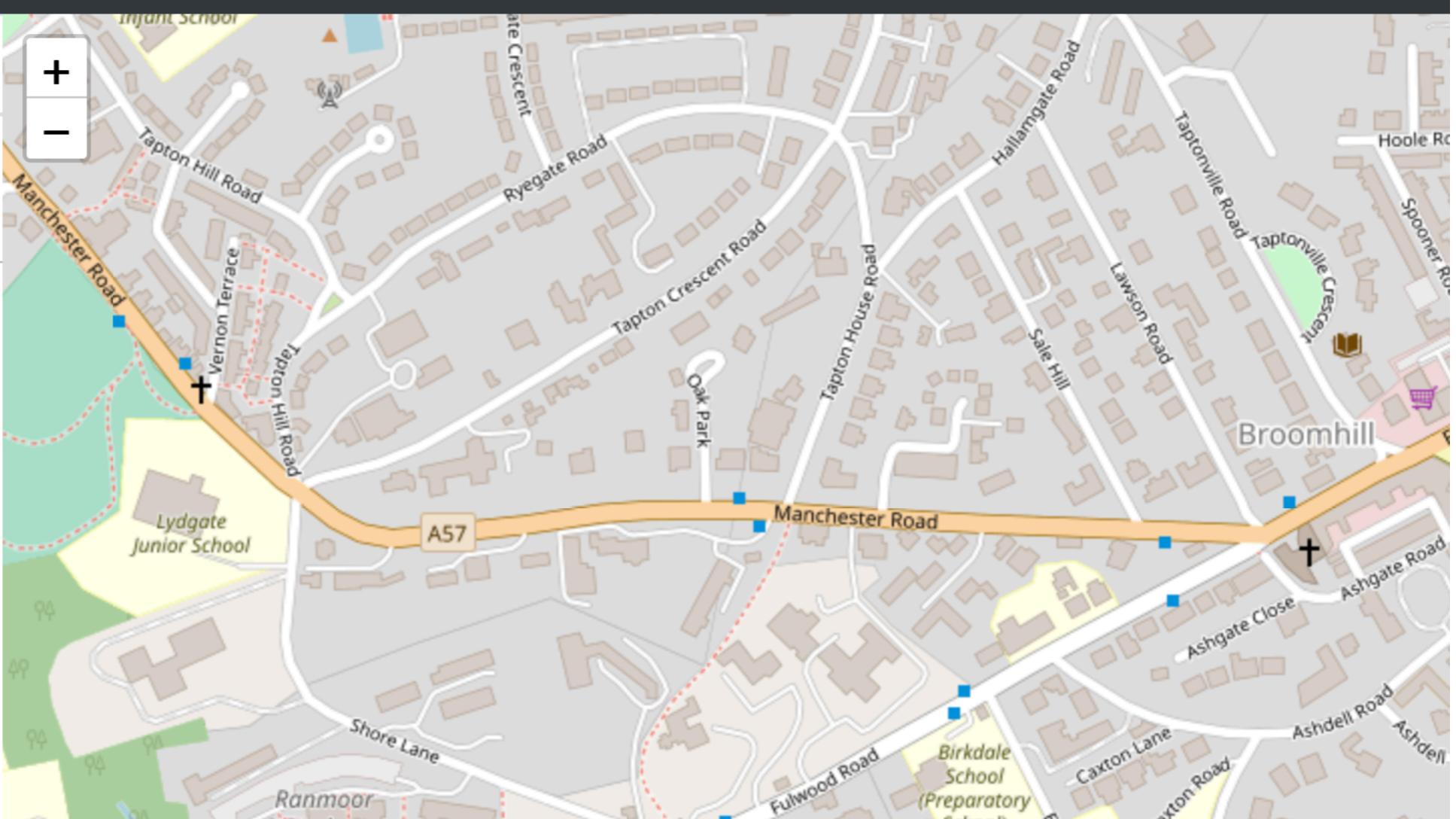
```
$ kubectl describe svc fleetman-webapp
Name:                   fleetman-webapp
Namespace:              default
Labels:                 <none>
Annotations:            kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"v1","kind":"Service","metadata":{"name":"fleetman-webapp","namespace":"default","labels":{},"annotations":{},"creationTimestamp":null}, "spec":{"type": "NodePort","ports":[{"name": "http","nodePort": 30080,"port": 80,"targetPort": "http"}],"selector":{"app": "webapp","release": "0-5"}}, "status": {"loadBalancer": {},"externalIPs": [], "internalIP": null,"ports": [{"port": 80,"nodePort": 30080,"protocol": "TCP","targetPort": "http"}]}}
Selector:              app=webapp,release=0-5
Type:                  NodePort
IP:                   10.107.11.35
Port:                 http  80/TCP
TargetPort:            80/TCP
NodePort:              http  30080/TCP
Endpoints:             172.17.0.5:80
Session Affinity:      None
External Traffic Policy: Cluster
Events:                <none>
```

- ▶ If we try it we can see that it is the correct release

Fleet Management System PROTOTYPE. Release 0.5

Name	Last seen	Speed mph
------	-----------	-----------

Live vehicle updates will appear here.
Once we've implemented it!



- ▶ What would happen if the pod - webapp-release-0-5 dies (for whatever reason)?
- ▶ Let's simulate that!
- ▶ `kubectl delete po webapp-release-0-5`

```
$ kubectl delete po webapp-release-0-5
pod "webapp-release-0-5" deleted
```

It's dead and it ain't coming back!

▶ kubectl get all

```
$ kubectl get all
NAME           READY   STATUS    RESTARTS   AGE
pod/queue      1/1     Running   0          40m
pod/webapp     1/1     Running   2          3d

NAME          I
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
service/fleetman-queue   NodePort    10.100.30.30  <none>       8161:30010/TCP  37m
service/fleetman-webapp  NodePort    10.107.11.35  <none>       80:30080/TCP   3d
service/kubernetes     ClusterIP   10.96.0.1     <none>       443/TCP        4d
```

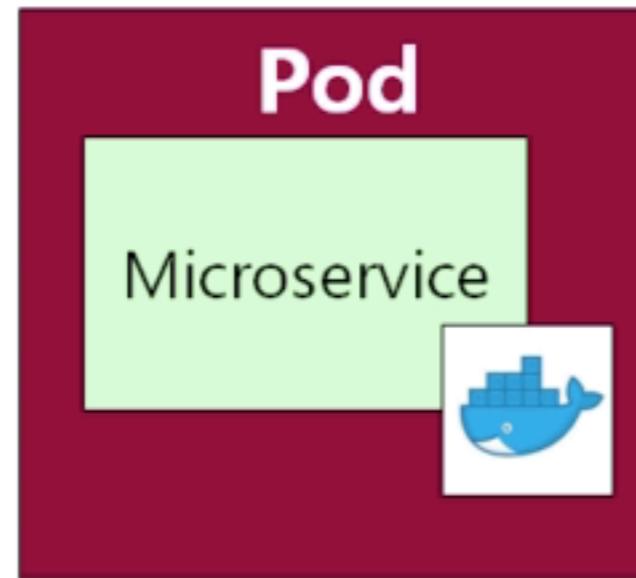
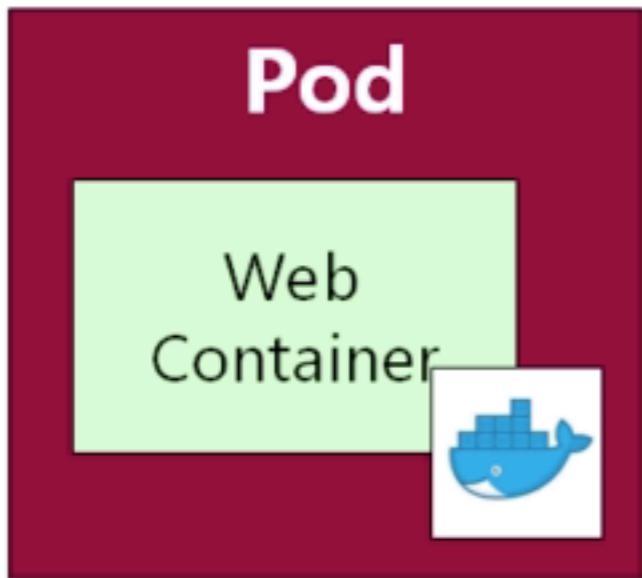
▶ If you go back to the browser

Safari Can't Connect to the Server

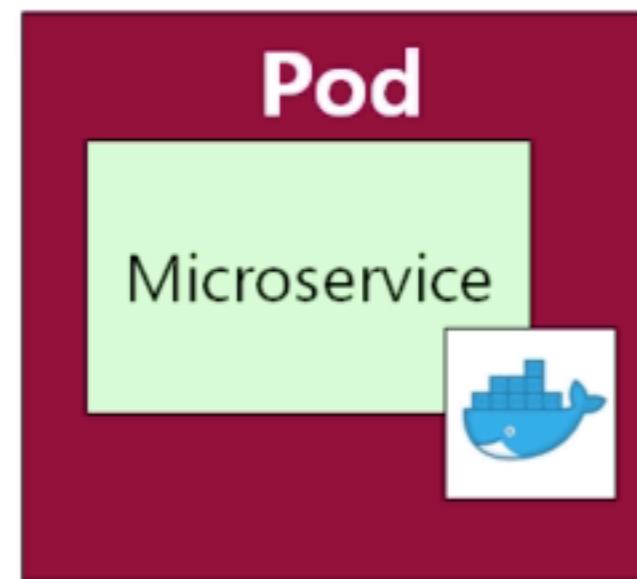
Safari can't open the page "192.168.99.100" because Safari can't connect to the server "192.168.99.100".

We don't want this to happen!

- ▶ Imagine that this happens at 3 o'clock of the morning....
- ▶ This is one of the reason we don't ususally deploy pods directly
- ▶ We should deploy a ReplicaSet to solve this problem



- ▶ A ReplicaSet is an extra piece of configuration we give to kubernetes



- We can specify how many instances of this pod we want to have at any time

- ▶ If the pod dies, for any reason, kubernetes creates another one
- ▶ This approach will be used for all our pods



- ▶ Go to the documentation of a ReplicaSet

<https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.16/#replicaset-v1-apps>

ReplicaSet v1 apps

[kubectl example](#)[curl example](#)

Group	Version	Kind
apps	v1	ReplicaSet

⚠ Warning:

In many cases it is recommended to create a [Deployment](#) instead of ReplicaSet.

▣ Other API versions of this object exist: [v1beta2](#) [v1beta1](#)

**on the
API page**

▶ YAML file for a ReplicaSet pod

```
apiVersion: extensions/v1beta1
kind: ReplicaSet
metadata:
  # Unique key of the ReplicaSet instance
  name: replicaset-example
spec:
  # 3 Pods should exist at all times.
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        # Run the nginx image
        - name: nginx
          image: nginx:1.10
```

it's outdated

**You must pay attention
for the group because
ReplicaSets are not from
the group “core”**

**definition of the pods - you don't
need a separate file for the pod
definition**

- ▶ Create a YAML text file for the ReplicaSet -
`pods.yaml`

```
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: webapp
5  spec:
6    selector:
7      matchLabels:
8        app: webapp
9    replicas: 1 ← if crashes we have a problem
10   template: # template for the pods
11     metadata:
12       labels:
13         app: webapp
14     spec:
15       containers:
16         - name: webapp
17           image: richardchesterwood/k8s-fleetman-webapp-angular:release0-5
18
```



- ▶ In this example we have:
 - ▶ Key - app
 - ▶ Value - web app

This is the way we connect ReplicaSets to Pods

- ▶ To apply this new definition, and because we don't have manual pods anymore is better to delete all pods
- ▶ `kubectl delete po --all`

```
$ kubectl delete po --all
pod "queue" deleted
pod "webapp-release-0" deleted
pod "webapp-release-0-5" deleted
```

▶ kubectl get all

\$ kubectl get all					
NAME	READY	STATUS	RESTARTS	AGE	
pod/queue	1/1	Terminating	0	1m	←
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP	1h
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP	3d
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d

- ▶ I was very quick - it was not terminated yet

▶ kubectl get all

\$ kubectl get all					
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP	1h
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP	3d
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d

- ▶ `kubectl apply -f pods.yaml`

```
$ kubectl apply -f pods.yaml
replicaset.apps "webapp" created
pod "queue" created
```

- ▶ `kubectl get all`

```
$ kubectl get all
NAME                 READY   STATUS    RESTARTS   AGE
pod/queue            1/1     Running   0          28s
pod/webapp-t9d81    1/1     Running   0          28s

NAME                           TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)           AGE
service/fleetman-queue         NodePort    10.100.30.30   <none>        8161:30010/TCP   1h
service/fleetman-webapp       NodePort    10.107.11.35   <none>        80:30080/TCP    3d
service/kubernetes            ClusterIP   10.96.0.1      <none>        443/TCP          4d

NAME            DESIRED   CURRENT   READY   AGE
replicaset.apps/webapp    1         1         1       29s
```

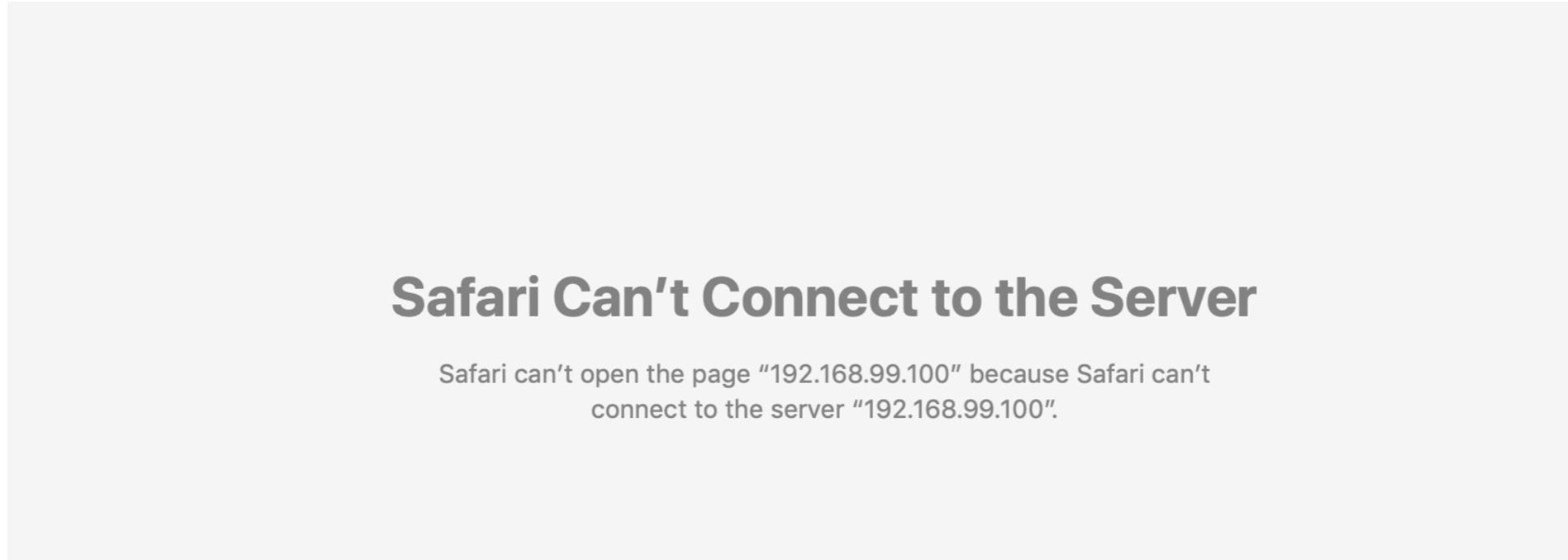


- ▶ **Desired** is what we want in terms of pods
- ▶ **Current** is what we have
- ▶ **Ready** is what pods are responding to requests

- ▶ We can explore the ReplicaSet
- ▶ `kubectl describe rs webapp`

```
$ kubectl describe rs webapp
Name:           webapp
Namespace:      default
Selector:       app=webapp
  tions:  kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"apps/v1","kind":"ReplicaSet","metadata":{"name":"webapp","namespace":"default"},"spec":{"replicas":1,"selector":{"matchLabels":{"app":"webapp"}}, "status":{"availableReplicas":1,"desiredReplicas":1,"fullyLabeledReplicas":1,"readyReplicas":1,"observedGeneration":1,"updatedReplicas":1}}}
  Replicas:  1 current / 1 desired
  Pods Status:  1 Running / 0 Waiting / 0 Succeeded / 0 Failed
  Pod Template:
    Labels:  app=webapp
  Containers:
    webapp:
      Image:      richardchesterwood/k8s-fleetman-webapp-angular:release0-5
      Port:       <none>
      Host Port: <none>
      Environment: <none>
      Mounts:     <none>
      Volumes:    <none>
  Events:
    Type  Reason          Age   From            Message
    ----  -----          ----  ----
    Normal SuccessfulCreate 3m    replicaset-controller  Created pod: webapp-t9d81
```

- ▶ If you go back to the browser



- ▶ It doesn't work... why?

```
! webapp-service.yaml
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: fleetman-webapp
5
6 spec:
7   selector:
8     app: webapp
9     release: "0-5" ←
10
11 ports:
12   - name: http
13     port: 80
14     nodePort: 30080
15
16 type: NodePort
17
```

you have to remove this line

▶ kubectl apply -f services.yaml

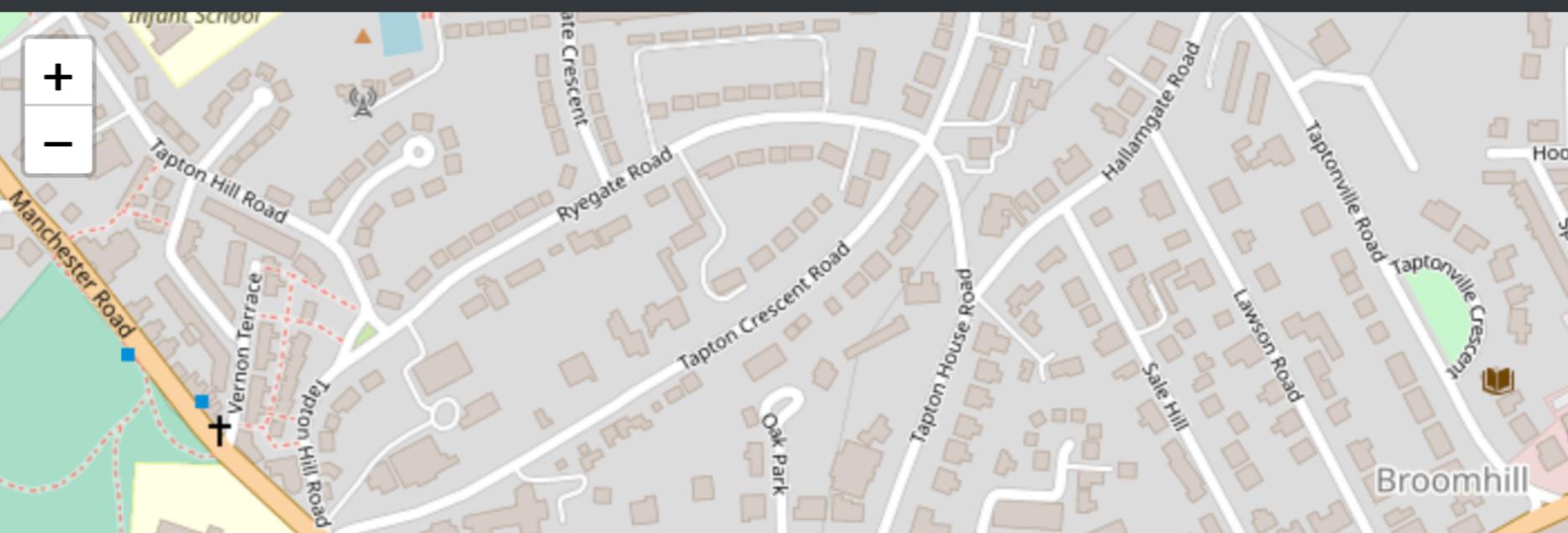
```
$ kubectl apply -f services.yaml
service "fleetman-webapp" configured
service "fleetman-queue" unchanged
```

- ▶ If we try it we can see that it is the correct release

Fleet Management System PROTOTYPE. Release 0.5

Name	Last seen	Speed mph
------	-----------	-----------

Live vehicle updates will appear here.
Once we've implemented it!



▶ kubectl get all

```
$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/queue	1/1	Running	0	9m
pod/webapp-t9d8l	1/1	Running	0	9m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP	1h
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP	3d
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/webapp	1	1	1	9m

*Let's simulate a pod
crashing*

- ▶ `kubectl delete po webapp-t9d81`

```
$ kubectl delete po webapp-t9d81
pod "webapp-t9d81" deleted
```

- ▶ `kubectl get all`

```
$ kubectl get all
NAME                 READY   STATUS            RESTARTS   AGE
pod/queue             1/1     Running           0          10m
pod/webapp-jlngh    0/1     ContainerCreating   0          4s
pod/webapp-t9d81    0/1     Terminating       0          10m   ←

NAME                           TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
service/fleetman-queue        NodePort    10.100.30.30   <none>        8161:30010/TCP  1h
service/fleetman-webapp       NodePort    10.107.11.35   <none>        80:30080/TCP   3d
service/kubernetes           ClusterIP  10.96.0.1      <none>        443/TCP         4d

NAME                DESIRED   CURRENT   READY   AGE
replicaset.apps/webapp 1         1         1       10m
```

▶ kubectl get all

```
$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/queue	1/1	Running	0	10m
pod/webapp-jlngh	1/1	Running	0	19s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP	1h
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP	3d
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/webapp	1	1	1	10m

- ▶ Lets analyse this again
- ▶ `kubectl describe rs webapp`

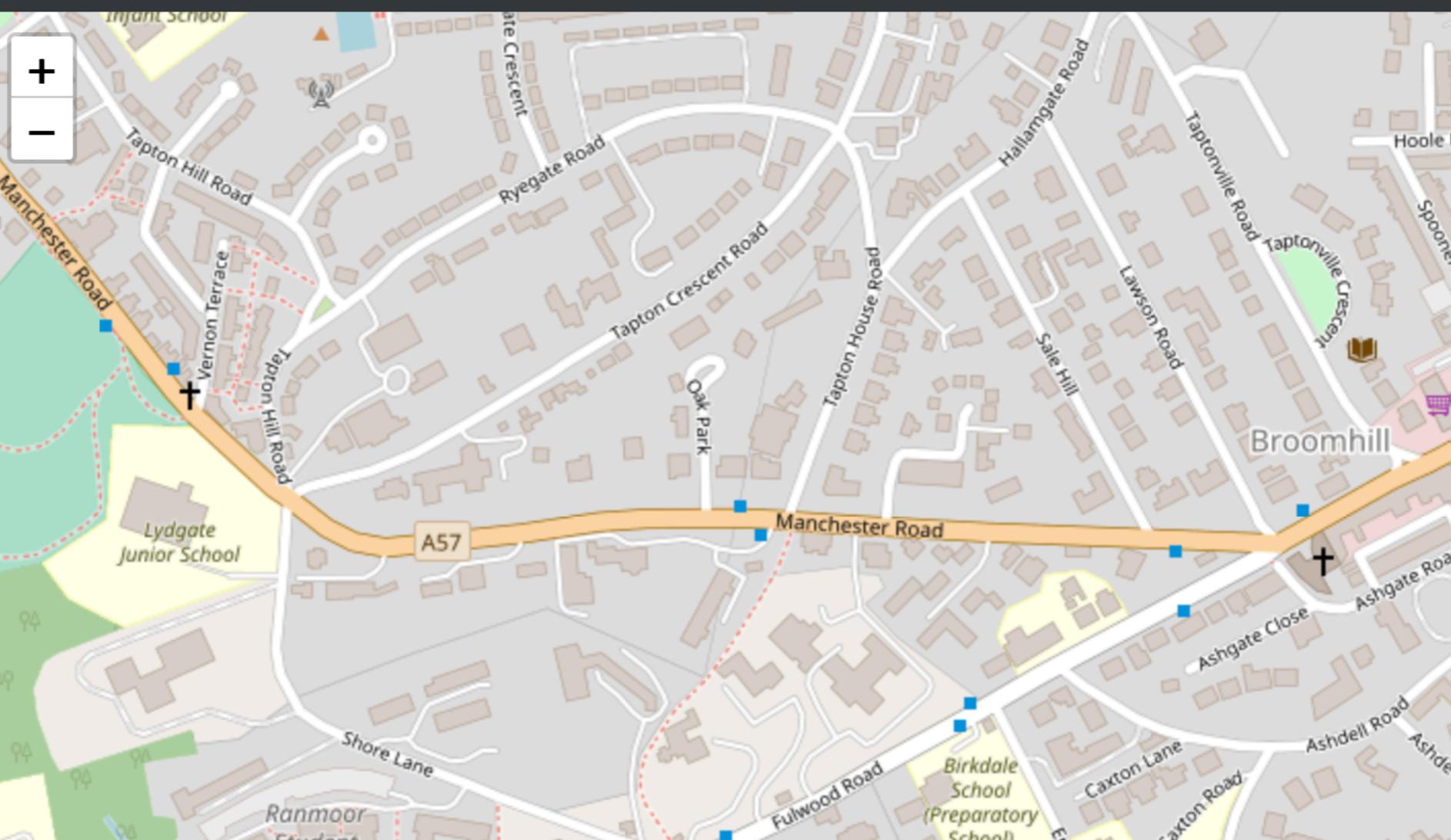
```
$ kubectl describe rs webapp
Name:           webapp
Namespace:      default
Selector:       app=webapp
Labels:         app=webapp
Annotations:   kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"apps/v1","kind":"ReplicaSet","metadata":{"name":"webapp","namespace":"default"},"spec":{"replicas":1,"selector":{"matchLabels":{"app":"webapp"}}, "template":{"metadata":{"labels":{"app":"webapp"}}, "spec":{}}}, controller-revision-hash: t9d81
Annotations:   kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"apps/v1","kind":"ReplicaSet","metadata":{"name":"webapp","namespace":"default"},"spec":{"replicas":1,"selector":{"matchLabels":{"app":"webapp"}}, "template":{"metadata":{"labels":{"app":"webapp"}}, "spec":{}}}, controller-revision-hash: j1ngh
Replicas:      1 current / 1 desired
Pods Status:   1 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=webapp
  Containers:
    webapp:
      Image:      richardchesterwood/k8s-fleetman-webapp-angular:release0-5
      Port:       <none>
      Host Port:  <none>
      Environment: <none>
      Mounts:     <none>
      Volumes:    <none>
Events:
  Type      Reason          Age    From            Message
  ----      -----          ----   ----
  Normal    SuccessfulCreate 10m    replicaset-controller  Created pod: webapp-t9d81
  Normal    SuccessfulCreate  35s   replicaset-controller  Created pod: webapp-j1ngh
```

- ▶ If you try it maybe you saw the page always online, but there was some downtime

Fleet Management System PROTOTYPE. Release 0.5

Name	Last seen	Speed mph
------	-----------	-----------

Live vehicle updates will appear here.
Once we've implemented it!



ReplicaSet

Replicas: 1

Pod

Web
Container



ReplicaSet

Replicas: 1

Pod

Microservice



- ▶ When the pod crashed there were no instances of that pod running!

ReplicaSet

Replicas: 2



ReplicaSet

Replicas: 1



- ▶ If one pod crashes, we have another

- # ▶ Change your file pods.yaml

▶ `kubectl apply -f pods.yaml`

```
$ kubectl apply -f pods.yaml
replicaset.apps "webapp" configured
pod "queue" unchanged
```

▶ `kubectl describe rs webapp`

```
$ kubectl describe rs webapp
Name:           webapp
Namespace:      default
Selector:       app=webapp
Labels:         app=webapp
Annotations:    kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"apps/v1","kind":"Replicaset","metadata":{"name":"webapp","namespace":"default"},"spec":{"replicas":2,"selector":{"matchLabels":{"app":"webapp"}}, "template":{"labels":{"app":"webapp"}, "spec":{}}}}
Replicas:      2 current / 2 desired
Pods Status:   2 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=webapp
  Containers:
    webapp:
      Image:      richardchesterwood/k8s-fleetman-webapp-angular:release0-5
      Port:       <none>
      Host Port:  <none>
      Environment: <none>
      Mounts:     <none>
      Volumes:    <none>
Events:
  Type      Reason          Age    From            Message
  ----      -----          ----   ----
  Normal    SuccessfulCreate 15m    replicaset-controller  Created pod: webapp-t9d81
  Normal    SuccessfulCreate  5m    replicaset-controller  Created pod: webapp-j1ngh
  Normal    SuccessfulCreate 10s    replicaset-controller  Created pod: webapp-4b5pc
```

▶ kubectl get all

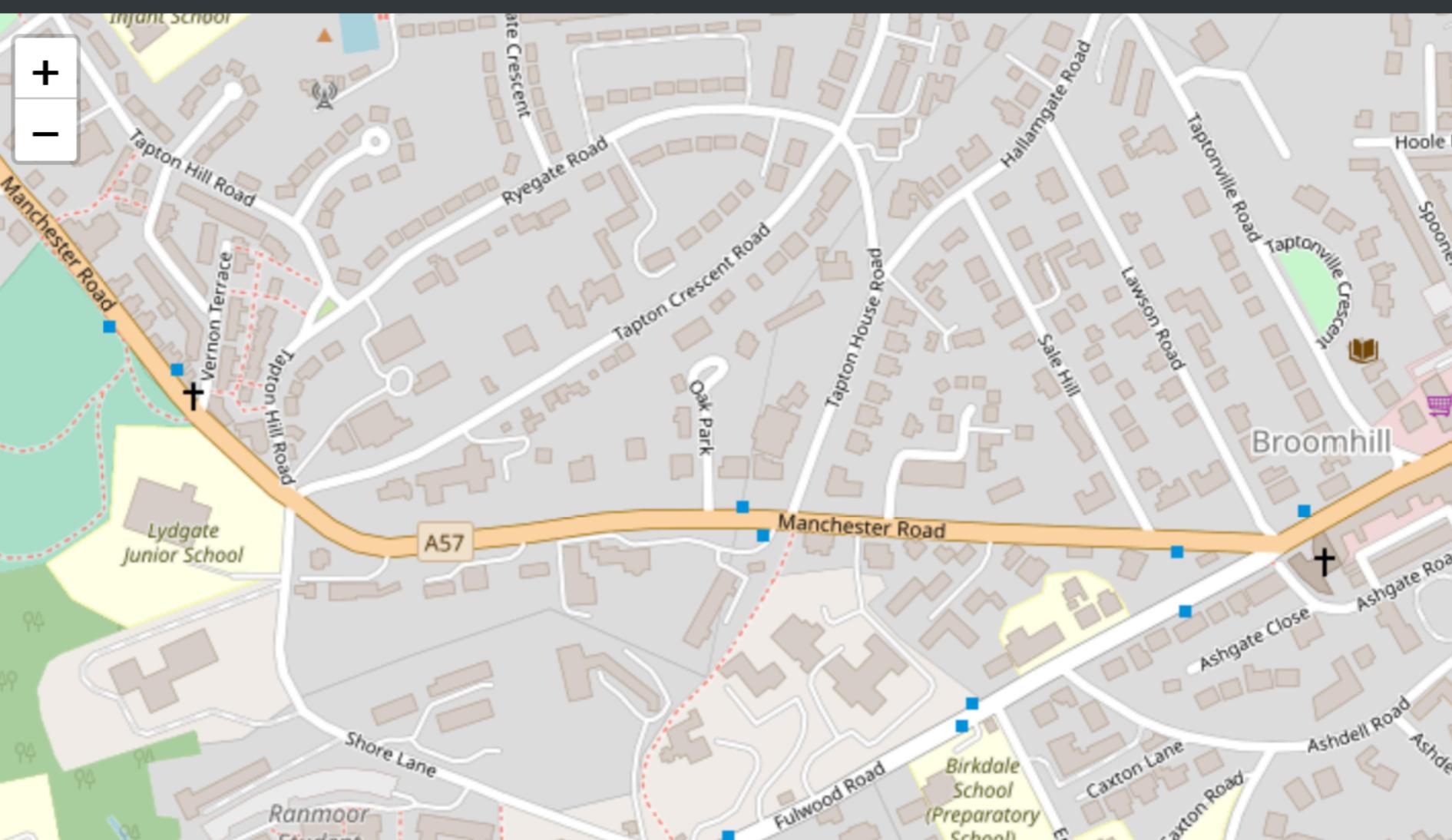
\$ kubectl get all					
NAME	READY	STATUS	RESTARTS	AGE	
pod/queue	1/1	Running	0	15m	
pod/webapp-4b5pc	1/1	Running	0	28s	
pod/webapp-j1ngh	1/1	Running	0	5m	
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP	1h
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP	3d
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d
NAME	DESIRED	CURRENT	READY	AGE	
replicaset.apps/webapp	2	2	2	15m	

- ▶ Delete one pod and try to get a not available error message

Fleet Management System PROTOTYPE. Release 0.5

Name	Last seen	Speed mph
------	-----------	-----------

Live vehicle updates will appear here.
Once we've implemented it!





kubernetes

Deployment

- ▶ With ReplicaSets we found that we don't want to work with Pods directly
- ▶ And Deployments could be better for our needs!

- ▶ Go to the documentation of a Deployment

<https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.16/#deployment-v1-apps>

Deployment v1 apps

[kubectl example](#)[curl example](#)

Group	Version	Kind
apps	v1	Deployment

▣ Other API versions of this object exist: v1beta2 v1beta1 v1beta1

ⓘ Appears In:

- DeploymentList [apps/v1]

**on the
API page**

- ▶ A Deployment is a more sophisticated form of a ReplicaSet
- ▶ It has one additional feature
 - ▶ with Deployments we have automatic rolling updates with zero downtime
 - ▶ We can do it manually (like we already done previously), but with deployments its automatic
 - ▶ We also have the possibility to do rollbacks - if something goes wrong

- ▶ The structure of the yaml file for the deployments is almost the same as the one for the ReplicaSet

▶ YAML file for a Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  # Unique key of the Deployment instance
  name: deployment-example
spec:
  # 3 Pods should exist at all times.
  replicas: 3
  template:
    metadata:
      labels:
        # Apply this label to pods and default
        # the Deployment label selector to this value
        app: nginx
    spec:
      containers:
        - name: nginx
          # Run this image
          image: nginx:1.10
```

it's almost the same

▶ kubectl get all

```
$ kubectl get all
NAME                 READY   STATUS    RESTARTS   AGE
pod/queue            1/1     Running   0          1h
pod/webapp-cbks9   1/1     Running   0          58m
pod/webapp-jlngh   1/1     Running   0          1h

NAME                  TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)      AGE
service/fleetman-queue  NodePort   10.100.30.30  <none>       8161:30010/TCP 2h
service/fleetman-webapp NodePort   10.107.11.35  <none>       80:30080/TCP   3d
service/kubernetes    ClusterIP  10.96.0.1     <none>       443/TCP     4d

NAME           DESIRED   CURRENT   READY   AGE
replicaset.apps/webapp  2         2         2       1h
```

▶ kubectl delete rs webapp

```
$ kubectl delete rs webapp
replicaset.extensions "webapp" deleted
```

▶ kubectl get all

\$ kubectl get all				
NAME	READY	STATUS	RESTARTS	AGE
pod/queue	1/1	Running	0	1h
pod/webapp-cbks9	0/1	Terminating	0	59m
pod/webapp-jlngh	0/1	Terminating	0	1h

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP	2h
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP	3d
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d

▶ kubectl get all

\$ kubectl get all				
NAME	READY	STATUS	RESTARTS	AGE
pod/queue	1/1	Running	0	1h
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP

- ▶ Create a YAML text file for the Deployment
 - `pods.yaml`

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: webapp
5  spec:
6    selector:
7      matchLabels:
8        app: webapp
9    replicas: 2
10   template: # template for the pods
11     metadata:
12       labels:
13         app: webapp
14     spec:
15       containers:
16         - name: webapp
17           image: richardchesterwood/k8s-fleetman-webapp-angular:release0
18
```



▶ `kubectl apply -f pods.yaml`

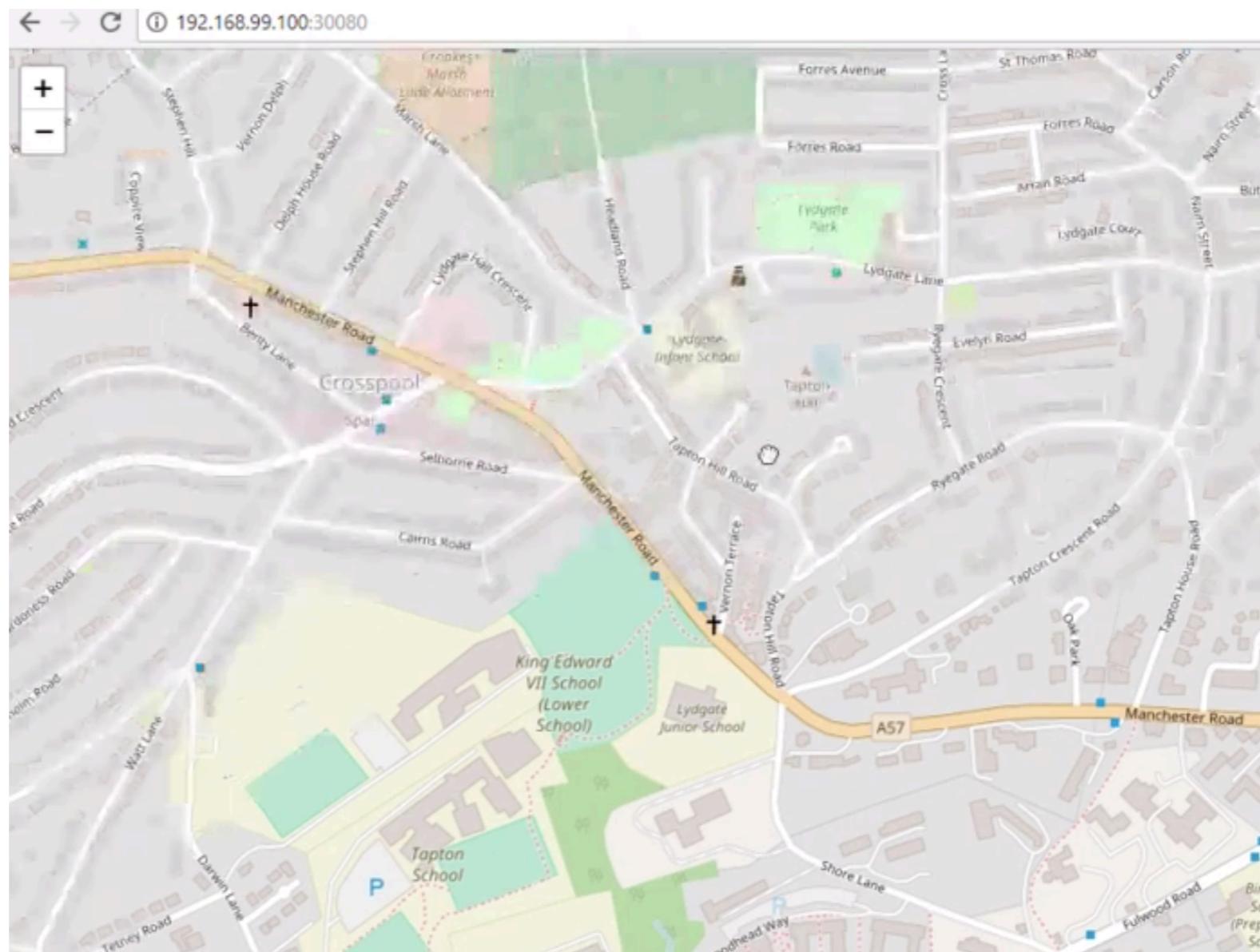
```
$ kubectl apply -f pods.yaml
deployment.apps "webapp" created
pod "queue" unchanged
```

▶ `kubectl get all`

NAME	READY	STATUS	RESTARTS	AGE	
pod/queue	1/1	Running	0	1h	
pod/webapp-746ff4c965-c9gn4	1/1	Running	0	10s	
pod/webapp-746ff4c965-vbts5	1/1	Running	0	10s	
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP	
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP	
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	
NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/webapp	2	2	2	2	10s
NAME	DESIRED	CURRENT	READY	AGE	
replicaset.apps/webapp-746ff4c965	2	2	2	10s	

- ▶ The deployment created a ReplicaSet
- ▶ The Deployment can be seen as an entity that manages the ReplicaSet
- ▶ For now on we don't need to work with the ReplicaSet directly we will work with the Deployment
- ▶ If you take a look, the Deployment created and manages the ReplicaSet and the ReplicaSet Created and manages the Pods

Let's try it now!



How can we do a rolling deployment without having to change the labels?

ReplicaSet

Replicas: 2

Pod

Web
Container
v1

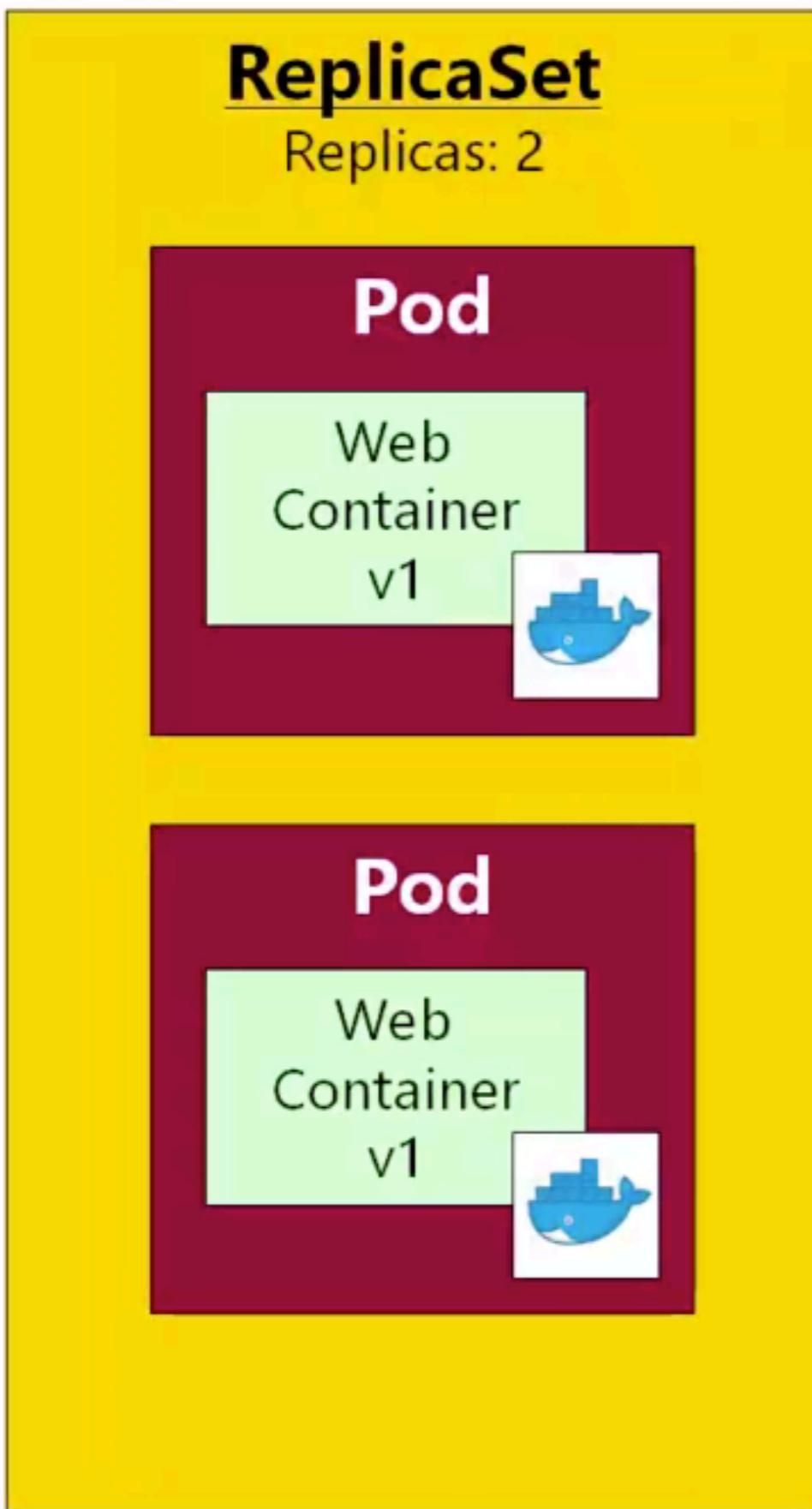


Pod

Web
Container
v1



- ▶ When we make a change to the image we are using in the deployment
- ▶ And we re-deploy that deployment
- ▶ Kubernetes will start a brand new ReplicaSet
- ▶ The Pods that are running on that new ReplicaSet will contain the updated image



ReplicaSet

Replicas: 2

Pod

Web
Container
v1



Pod

Web
Container
v1



ReplicaSet

Replicas: 2

Pod

Web
Container
v2



Pod

Web
Container
v2



ReplicaSet

Replicas: 2

Pod

Web
Container
v1



Pod

Web
Container
v1



ReplicaSet

Replicas: 2

Pod

Web
Container
v2



Pod

Web
Container
v2



ReplicaSet

Replicas: 2

Pod

Web
Container
v1



Pod

Web
Container
v1



ReplicaSet

Replicas: 2

Pod

Web
Container
v2



Pod

Web
Container
v2



- ▶ When the new Pods are responding to requests
- ▶ The required number of replicas in the old ReplicaSet will be switch to 0
- ▶ That means that the old Pods will be stoped

ReplicaSet

Replicas: 2

ReplicaSet

Replicas: 2

Pod

Web
Container
v2



Pod

Web
Container
v2



ReplicaSet

Replicas: 2

Pod

Web
Container
v1



Pod

Web
Container
v1



ReplicaSet

Replicas: 2

Pod

Web
Container
v2



Pod

Web
Container
v2



ReplicaSet

Replicas: 2

Pod

Web
Container
v1



Pod

Web
Container
v1



ReplicaSet

Replicas: 2

Pod

Web
Container
v2



Pod

Web
Container
v2



ReplicaSet

Replicas: 2

Pod

Web
Container
v1



Pod

Web
Container
v1

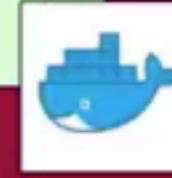


ReplicaSet

Replicas: 2

Pod

Web
Container
v2



Pod

Web
Container
v2



ReplicaSet

Replicas: 0

ReplicaSet

Replicas: 2

Pod

Web
Container
v2



Pod

Web
Container
v2



- ▶ The old ReplicaSet will continue to exist but with 0 replicas
- ▶ If something goes wrong with the new version we can go back to the old one

- ▶ This is all based on the container image
- ▶ The container image must be changed in order for this process to work

Let's do a rolling deployment for the release0-5

► pods.yaml

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    |  name: webapp
5  spec:
6    |  selector:
7      |    matchLabels:
8        |          app: webapp
9  replicas: 2
10 template: # template for the pods
11   |  metadata:
12     |    labels:
13       |      app: webapp
14   |  spec:
15     |    containers:
16       - name: webapp
17         |      image: richardchesterwood/k8s-fleetman-webapp-angular:release0-5
18
```

- ▶ Before applying this deploy let's tune it a little bit

DeploymentSpec v1 apps

i Appears In:

- Deployment [apps/v1]

Field	Description
<code>minReadySeconds</code> <i>integer</i>	Minimum number of seconds for which a newly created pod should be ready with Defaults to 0 (pod will be considered available as soon as it is ready)
<code>paused</code> <i>boolean</i>	Indicates that the deployment is paused.
<code>progressDeadlineSeconds</code> <i>integer</i>	The maximum time in seconds for a deployment to make progress before it is considered failed. If the deployment fails to make progress before the deadline, the deployment will be failed. This will limit the time spent on process failed deployments and a condition with a ProgressDeadlineExceeded reason will not be estimated during the time a deployment is paused. Defaults to 600s.
<code>replicas</code>	Number of desired pods. This is a pointer to distinguish between explicit zero and



► pods.yaml

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4  |   name: webapp
5  spec:
6  |   minReadySeconds: 30 ←
7  |   selector:
8  |   |   matchLabels:
9  |   |   |   app: webapp
10 |   replicas: 2
11 |   template: # template for the pods
12 |   |   metadata:
13 |   |   |   labels:
14 |   |   |   |   app: webapp
15 |   |   spec:
16 |   |   |   containers:
17 |   |   |   - name: webapp
18 |   |   |   |   image: richardchesterwood/k8s-fleetman-webapp-angular:release0-5
19 |
```

▶ `kubectl apply -f pods.yaml`

```
$ kubectl apply -f pods.yaml
deployment.apps "webapp" configured
pod "queue" unchanged
```

▶ `kubectl get all`

```
$ kubectl get all
NAME                           READY   STATUS    RESTARTS   AGE
pod/queue                       1/1     Running   0          1h
pod/webapp-746ff4c965-c9gn4   1/1     Running   0          16m
pod/webapp-746ff4c965-vbts5   1/1     Running   0          16m
pod/webapp-ccb5c74c9-rrn18    1/1     Running   0          4s

NAME                  TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)      AGE
service/fleetman-queue  NodePort    10.100.30.30  <none>       8161:30010/TCP 3h
service/fleetman-webapp NodePort    10.107.11.35  <none>       80:30080/TCP   3d
service/kubernetes       ClusterIP  10.96.0.1     <none>       443/TCP     4d

NAME            DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
deployment.apps/webapp  2         3         1           2          16m

NAME            DESIRED  CURRENT  READY   AGE
replicaset.apps/webapp-746ff4c965  2         2         2       16m
replicaset.apps/webapp-ccb5c74c9   1         1         1       4s
```



▶ kubectl get all

```
$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/queue	1/1	Running	0	1h
pod/webapp-746ff4c965-c9gn4	1/1	Running	0	16m
pod/webapp-746ff4c965-vbts5	1/1	Running	0	16m
pod/webapp-ccb5c74c9-rrn18	1/1	Running	0	25s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP	3h
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP	3d
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/webapp	2	3	1	2	16m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/webapp-746ff4c965	2	2	2	16m
replicaset.apps/webapp-ccb5c74c9	1	1	1	25s

▶ kubectl get all

```
$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/queue	1/1	Running	0	1h
pod/webapp-746ff4c965-c9gn4	1/1	Running	0	17m
pod/webapp-ccb5c74c9-rrn18	1/1	Running	0	50s
pod/webapp-ccb5c74c9-zqkps	1/1	Running	0	18s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP	3h
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP	3d
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/webapp	2	3	2	2	17m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/webapp-746ff4c965	1	1	1	17m
replicaset.apps/webapp-ccb5c74c9	2	2	2	50s

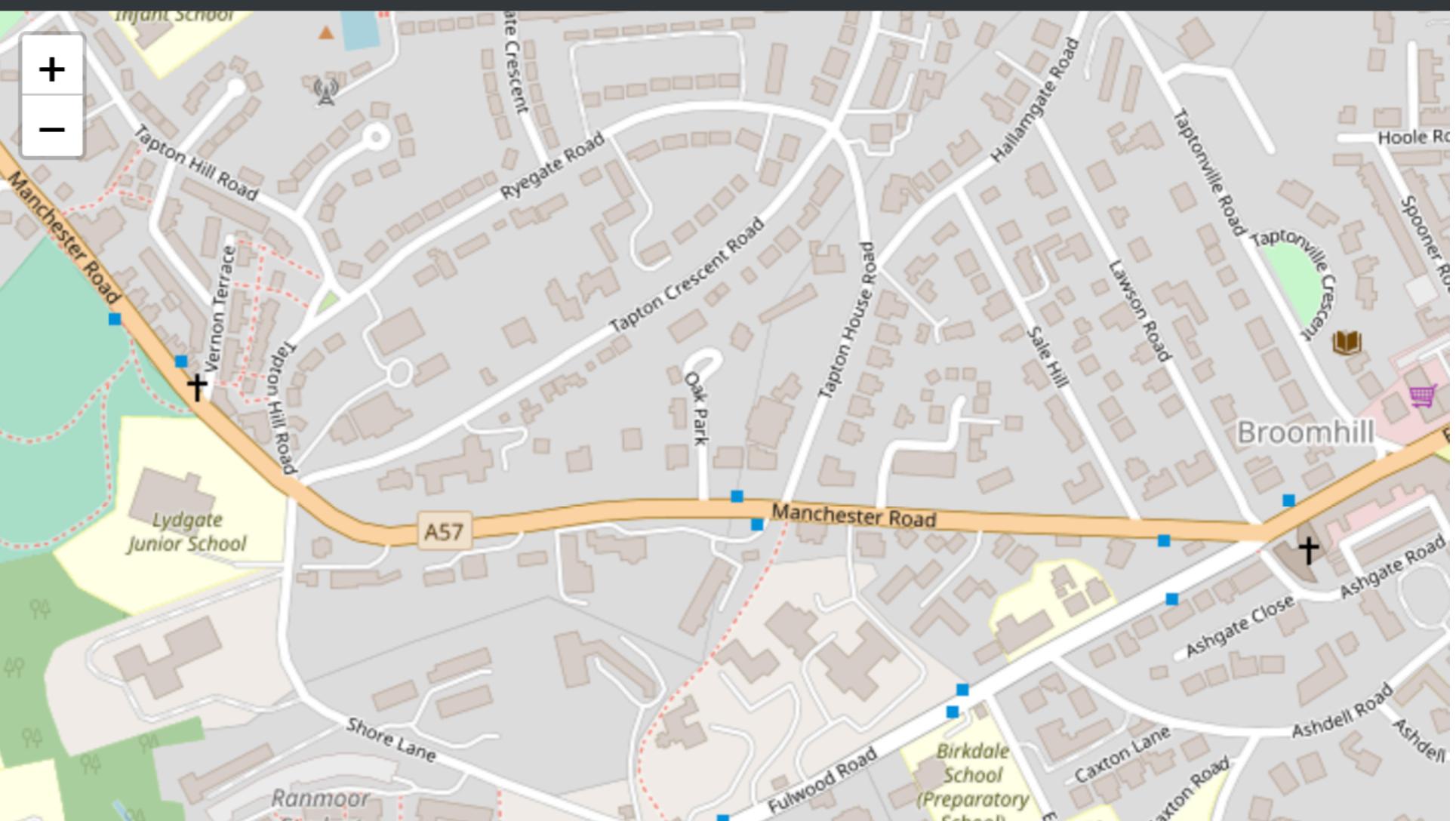


- ▶ Keep refreshing the browser till the new version appears!

Fleet Management System PROTOTYPE. Release 0.5

Name	Last seen	Speed mph
------	-----------	-----------

Live vehicle updates will appear here.
Once we've implemented it!



▶ kubectl get all

\$ kubectl get all					
NAME	READY	STATUS	RESTARTS	AGE	
pod/queue	1/1	Running	0	1h	
pod/webapp-ccb5c74c9-rrn18	1/1	Running	0	1m	
pod/webapp-ccb5c74c9-zqkps	1/1	Running	0	46s	
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161: 30010/TCP	3h
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80: 30080/TCP	3d
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d
NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/webapp	2	2	2	2	17m
NAME	DESIRED	CURRENT	READY	AGE	
replicaset.apps/webapp-746ff4c965	1	0	0	17m	←
replicaset.apps/webapp-ccb5c74c9	2	2	2	1m	

Let's work with rollout's

► pods.yaml

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: webapp
5  spec:
6    minReadySeconds: 30
7    selector:
8      matchLabels:
9        app: webapp
10   replicas: 2
11   template: # template for the pods
12     metadata:
13       labels:
14         app: webapp
15     spec:
16       containers:
17         - name: webapp
18           image: richardchesterwood/k8s-fleetman-webapp-angular:release0
19
```



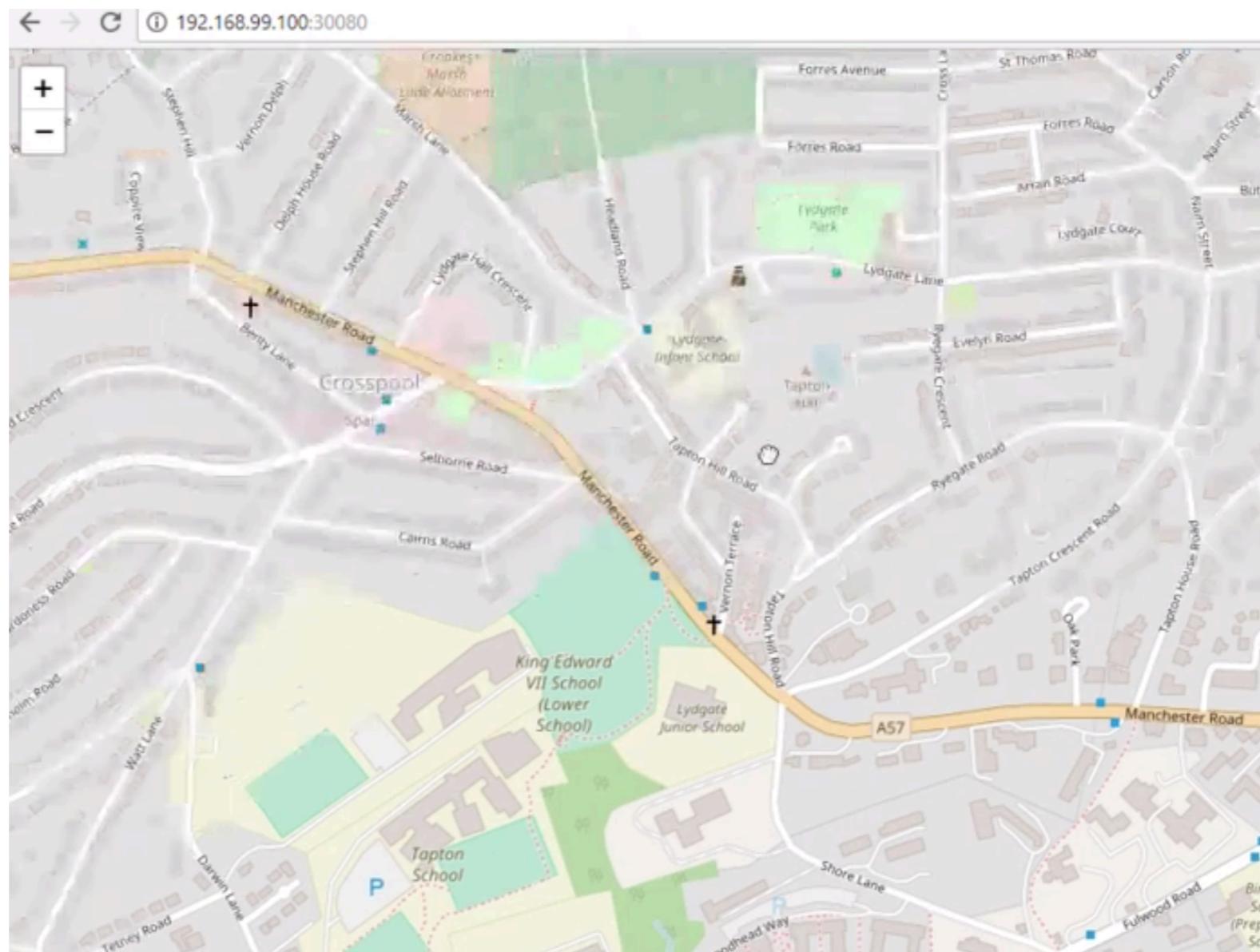
- ▶ `kubectl apply -f pods.yaml`

```
$ kubectl apply -f pods.yaml
deployment.apps "webapp" configured
pod "queue" unchanged
```

- ▶ `kubectl rollout status deploy webapp`

```
$ kubectl rollout status deploy webapp
Waiting for rollout to finish: 1 out of 2 new replicas have been updated...
Waiting for rollout to finish: 1 out of 2 new replicas have been updated...
Waiting for rollout to finish: 1 out of 2 new replicas have been updated...
Waiting for rollout to finish: 1 old replicas are pending termination...
Waiting for rollout to finish: 1 old replicas are pending termination...
Waiting for rollout to finish: 1 old replicas are pending termination...
deployment "webapp" successfully rolled out
```

Let's try it now!



- ▶ `kubectl rollout history deploy webapp`

```
$ kubectl rollout history deploy webapp
deployments "webapp"
REVISION  CHANGE-CAUSE
2          <none>
3          <none>
```

- ▶ Revision 3 is the current live
- ▶ Revision 2 is the last one (the release0-5)

How to go back to the old version?

- ▶ `kubectl rollout undo deploy webapp`

```
$ kubectl rollout undo deploy webapp  
deployment.apps "webapp"
```

- ▶ You could specify the revision you want to go back to with the following parameter:
`--to-revision=2`

- ▶ `kubectl rollout history deploy webapp`

```
$ kubectl rollout history deploy webapp
deployments "webapp"
REVISION  CHANGE-CAUSE
3          <none>
4          <none> ←
```

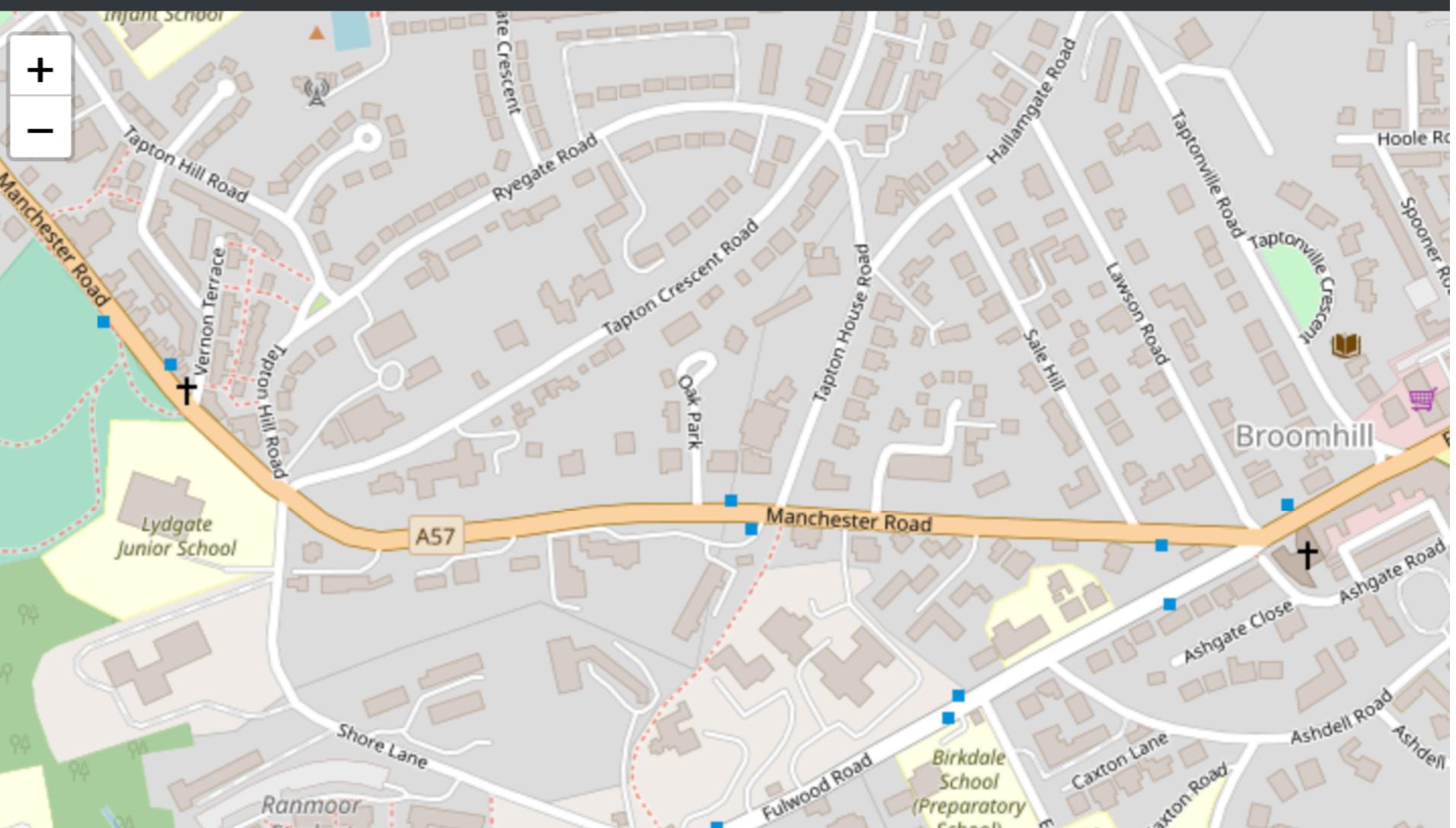
- ▶ Revision 4 is the same as the old Revision 2

Let's try it now!

Fleet Management System PROTOTYPE. Release 0.5

Name	Last seen	Speed mph
------	-----------	-----------

Live vehicle updates will appear here.
Once we've implemented it!



▶ kubectl get all

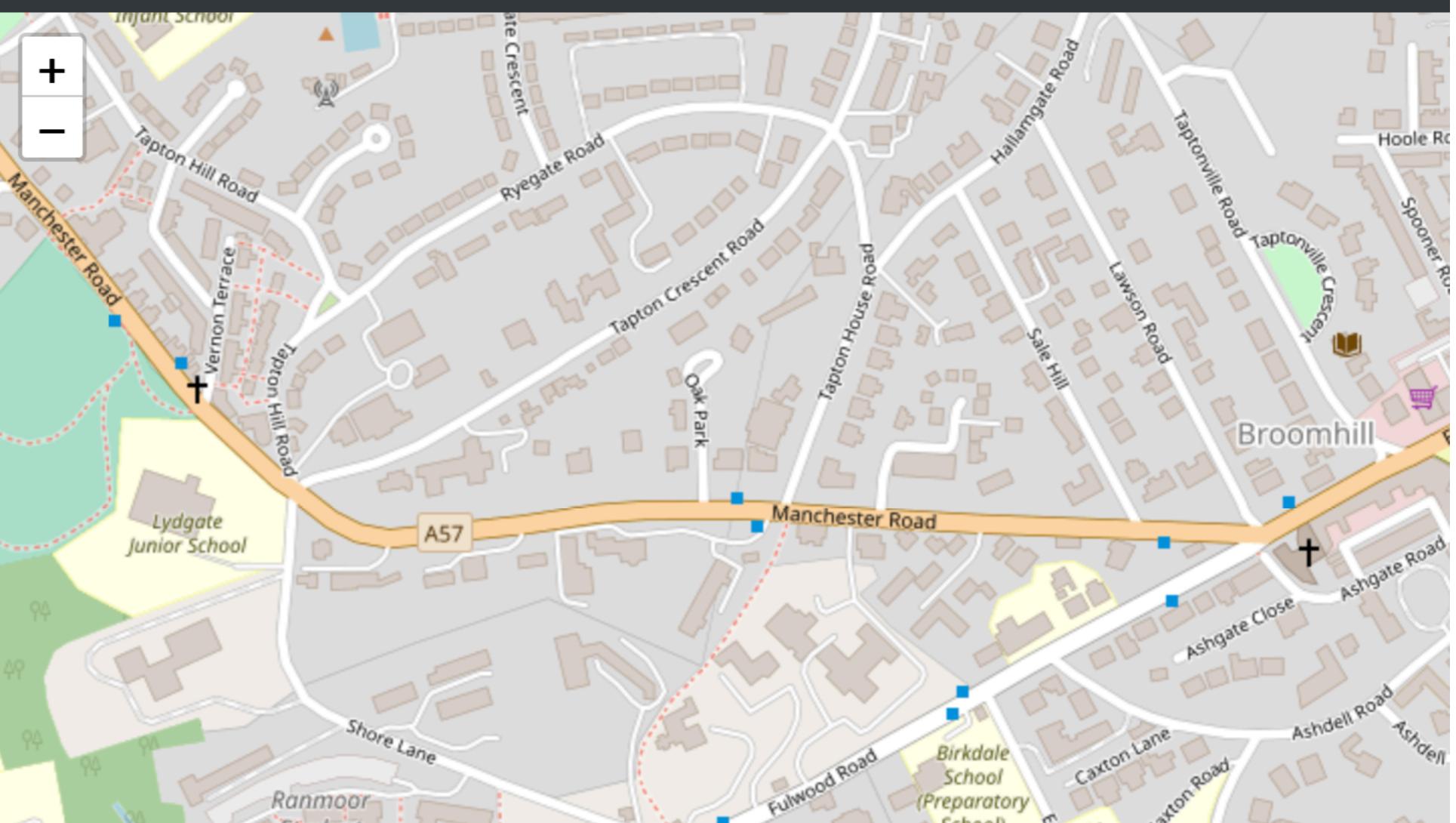
\$ kubectl get all					
NAME	READY	STATUS	RESTARTS	AGE	
pod/queue	1/1	Running	0	1h	
pod/webapp-ccb5c74c9-9m6w7	1/1	Running	0	1m	
pod/webapp-ccb5c74c9-1cqrn	1/1	Running	0	1m	
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP	3h
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP	3d
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d
NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/webapp	2	2	2	2	30m
NAME	DESIRED	CURRENT	READY	AGE	
replicaset.apps/webapp-746ff4c965	0	0	0	30m	
replicaset.apps/webapp-ccb5c74c9	2	2	2	14m	

Let's try it now!

Fleet Management System PROTOTYPE. Release 0.5

Name	Last seen	Speed mph
------	-----------	-----------

Live vehicle updates will appear here.
Once we've implemented it!



- ▶ Attention that this last version doesn't match your last YAML file
- ▶ These rollback are to use only in emergencies!

*Let's try to deploy a
damaged image*

► pods.yaml

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: webapp
5  spec:
6    #minReadySeconds: 30 ←
7    selector:
8      matchLabels:
9        app: webapp
10   replicas: 2
11   template: # template for the pods
12     metadata:
13       labels:
14         app: webapp
15     spec:
16       containers:
17         - name: webapp
18           image: richardchesterwood/k8s-fleetman-webapp-angular:releaqweqweqwse0
19
```

**the pull of this image it is
going to fail**



▶ kubectl get all

\$ kubectl get all						
NAME	READY	STATUS	RESTARTS	AGE	PORT(S)	AGE
pod/queue	1/1	Running	0	1h	8161:30010/TCP	3h
pod/webapp-ccb5c74c9-9m6w7	1/1	Running	0	5m	80:30080/TCP	3d
pod/webapp-ccb5c74c9-1cqrn	1/1	Running	0	5m	443/TCP	4d
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE	AGE
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP	3h	
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP		
service/kubernetes	ClusterIP	10.96.0.1	<none>			
NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE	AGE
deployment.apps/webapp	2	2	2	2	34m	
NAME	I	DESIRED	CURRENT	READY	AGE	AGE
replicaset.apps/webapp-746ff4c965	0	0	0	0	34m	
replicaset.apps/webapp-ccb5c74c9	2	2	2	2	18m	

- ▶ The current running ReplicaSet - ccb
- ▶ Scaled down ReplicaSet - 746

▶ `kubectl apply -f pods.yaml`

```
$ kubectl apply -f pods.yaml
deployment.apps "webapp" configured
pod "queue" unchanged
```

▶ `kubectl get all`

NAME	READY	STATUS	RESTARTS	AGE
pod/queue	1/1	Running	0	1h
pod/webapp-64cd5c487f-smw25	0/1	ContainerCreating	0	5s
pod/webapp-ccb5c74c9-9m6w7	1/1	Running	0	6m
pod/webapp-ccb5c74c9-1cqrm	1/1	Running	0	5m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP	3h
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP	3d
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/webapp	2	3	1	2	35m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/webapp-64cd5c487f	1	1	0	5s
replicaset.apps/webapp-746ff4c965	0	0	0	35m
replicaset.apps/webapp-ccb5c74c9	2	2	2	18m

▶ kubectl get all

\$ kubectl get all					
NAME	READY	STATUS	RESTARTS	AGE	
pod/queue	1/1	Running	0	1h	
pod/webapp-64cd5c487f-smw25	0/1	ImagePullBackoff	0	50s	←
pod/webapp-ccb5c74c9-9m6w7	1/1	Running	0	7m	
pod/webapp-ccb5c74c9-1cqrm	1/1	Running	0	6m	
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP	3h
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP	3d
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d
NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/webapp	2	3	1	2	36m
NAME	DESIRED	CURRENT	READY	AGE	
replicaset.apps/webapp-64cd5c487f	1	1	0	50s	←
replicaset.apps/webapp-746ff4c965	0	0	0	36m	
replicaset.apps/webapp-ccb5c74c9	2	2	2	19m	

- ▶ Kubernetes will continue to try to download the image because it could be a network problem

▶ **kubectl describe pod****webapp-64cd5c487f-smw25**

```
Reason:           ImagePullBackOff
Ready:            False
Restart Count:   0
Environment:     <none>
Mounts:
    /var/run/secrets/kubernetes.io/serviceaccount from default-token-wpbqh (ro)
Conditions:
  Type        Status
  Initialized  True
  Ready        False
  PodScheduled True
Volumes:
  default-token-wpbqh:
    Type:          Secret (a volume populated by a Secret)
    SecretName:   default-token-wpbqh
    Optional:     false
  QoS Class:    BestEffort
  Node-Selectors: <none>
  Tolerations:   node.kubernetes.io/not-ready:NoExecute for 300s
                  node.kubernetes.io/unreachable:NoExecute for 300s
Events:
  Type  Reason          Age   From            Message
  ----  ----          ----  ----            -----
  Normal Scheduled      2m    default-scheduler  Successfully assigned webapp-64cd5c487f-smw25 to minikube
  Normal SuccessfulMountVolume 2m    kubelet, minikube  MountVolume.SetUp succeeded for volume "default-token-wpbqh"
  Normal Pulling        1m (x4 over 2m)  kubelet, minikube  pulling image "richardcheung:releASDFASDFASFSAase0"
  Warning Failed        58s (x4 over 2m)  kubelet, minikube  Failed to pull image "richardcheung:releASDFASDFASFSAase0": rpc error: code = Unknown desc = Error response from daemon: 8s-fleetman-webapp-angular:releASDFASDFASFSAase0 not found
  Warning Failed        58s (x4 over 2m)  kubelet, minikube  Error: ErrImagePull
  Normal BackOff       30s (x6 over 2m)  kubelet, minikube  Back-off pulling image "richardcheung:releASDFASDFASFSAase0"
```

- ▶ The image will never be pulled off but the system keeps healthy
- ▶ We keep using the same ReplicaSet

```
$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/queue	1/1	Running	0	1h
pod/webapp-64cd5c487f-smw25	0/1	ContainerCreating	0	5s
pod/webapp-ccb5c74c9-9m6w7	1/1	Running	0	6m
pod/webapp-ccb5c74c9-1cqrm	1/1	Running	0	5m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP	3h
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP	3d
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/webapp	2	3	1	2	35m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/webapp-64cd5c487f	1	1	0	5s
replicaset.apps/webapp-746ff4c965	0	0	0	35m
replicaset.apps/webapp-ccb5c74c9	2	2	2	18m

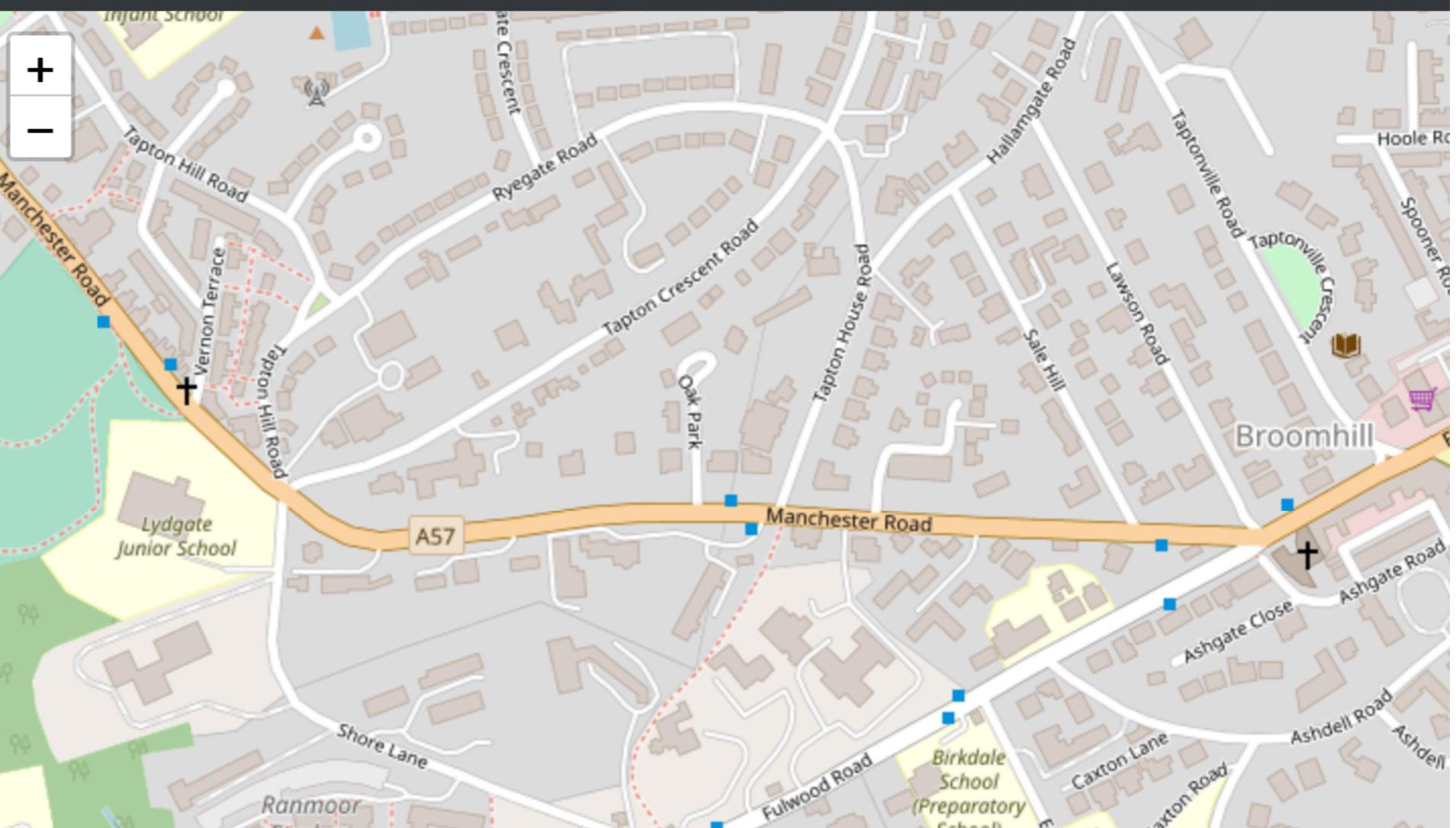


You can verify

Fleet Management System PROTOTYPE. Release 0.5

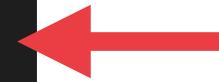
Name	Last seen	Speed mph
------	-----------	-----------

Live vehicle updates will appear here.
Once we've implemented it!



Let's fix the problem

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4  |  name: webapp
5  spec:
6  |  #minReadySeconds: 30
7  |  selector:
8  |  |  matchLabels:
9  |  |  |  app: webapp
10 |  |  replicas: 2
11 |  |  template: # template for the pods
12 |  |  metadata:
13 |  |  |  labels:
14 |  |  |  |  app: webapp
15 |  |  spec:
16 |  |  |  containers:
17 |  |  |  - name: webapp
18 |  |  |  |  image: richardchesterwood/k8s-fleetman-webapp-angular:release0-5
19
```



▶ `kubectl apply -f pods.yaml`

```
$ kubectl apply -f pods.yaml
deployment.apps "webapp" configured
pod "queue" unchanged
```

▶ `kubectl get all`

NAME					
	READY	STATUS	RESTARTS	AGE	
pod/queue	1/1	Running	0	1h	
pod/webapp-64cd5c487f-smw25	0/1	Terminating	0	4m	←
pod/webapp-ccb5c74c9-9m6w7	1/1	Running	0	11m	
pod/webapp-ccb5c74c9-1cqrnm	1/1	Running	0	10m	
NAME					
	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-queue	NodePort	10.100.30.30	<none>	8161:30010/TCP	3h
service/fleetman-webapp	NodePort	10.107.11.35	<none>	80:30080/TCP	3d
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d
NAME					
	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/webapp	2	2	2	2	40m
NAME					
	DESIRED	CURRENT	READY	AGE	
replicaset.apps/webapp-64cd5c487f	0	0	0	4m	
replicaset.apps/webapp-746ff4c965	0	0	0	40m	
replicaset.apps/webapp-ccb5c74c9	2	2	2	23m	←

- ▶ Kubernetes detected we already had that ReplicaSet, so it was quick
- ▶ If you continue to do kubectl get all you will see the broken ReplicaSet will not be removed, that because kubernetes stores the last 10 rollouts
- ▶ If you want to delete you must do it manually

P.PORTO

Polytechnic of
Porto

ESTG - School of
Management and Technology