LONDON
INFRASTRUCTURE
AS CODE

Introduction to Containers & Kubernetes

Meetup guide

H-EAC: Version 2020-02-02

Table of Contents

What are Linux Containers:

Linux containers, in short, contain applications in a way that keep them isolated from the host system that they run on. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package. And they are designed to make it easier to provide a consistent experience as developers and system administrators move code from development environments into production in a fast and replicable way.

Source: https://opensource.com/resources/what-are-linux-containers

Build a container image

Get started by creating a static (basic) website on a container, running it locally on your vagrant VM and viewing it on a web browser.

For this demo I'm using windows PC with the `ubuntu/xenial64` vagrant VM running in Virtualbox.

Pre-requisites to install on your VM using APT or YUM: **docker.io**
Install kubectl on your PC (I used version v1.15): https://kubernetes.io/docs/tasks/tools/install-kubectl/

On your server, go to the directory holding the website file:

Example:

```
vagrant@devops1:~$ mkdir src
vagrant@devops1:~$ cd src
vagrant@devops1:~/src$ vi index.html

<!doctype html>
<title>LondonIAC meetup - Site Maintenance</title>
<style>
  body { text-align: center; padding: 150px; }
  h1 { font-size: 50px; }
  body { font: 20px Helvetica, sans-serif; color: #333; }
  article { display: block; text-align: left; width: 650px; margin: 0 auto; }
  a { color: #dc8100; text-decoration: none; }
  a:hover { color: #333; text-decoration: none; }
</style>

<article>
    <h1>We&rsquo;ll be back soon!</h1>
    <div>
        <p>Sorry for the inconvenience but we&rsquo;re performing some maintenance at the moment. If
you need to you can always <a href="mailto:#">contact us</a>, otherwise we&rsquo;ll be back online
shortly!</p>
    <img src="https://marcelorjava.files.wordpress.com/2014/04/dilbert.gif" alt="Dilbert">
        <p>&mdash; The Team</p>
    </div>
    <p>
</p>
</article>

vagrant@devops1:~/src$ vi Dockerfile

      FROM nginx:alpine
      COPY . /usr/share/nginx/html
      EXPOSE 80

vagrant@devops1:~/src$ sudo docker build -t meetup-app .
Sending build context to Docker daemon  3.584kB
Step 1/3 : FROM nginx:alpine
 ---> ea1193fd3dde
Step 2/3 : COPY . /usr/share/nginx/html
 ---> 33bab77e254d
Step 3/3 : EXPOSE 80
```

```
 ---> Running in 3c9cc1e254b2
Removing intermediate container 3c9cc1e254b2
 ---> 1e743fba89b1
Successfully built 1e743fba89b1
Successfully tagged meetup-app:latest
```

Check the image has been created:

```
vagrant@devops1:~/src$ sudo docker images
REPOSITORY           TAG                  IMAGE ID            CREATED             SIZE
meetup-app            latest               1e743fba89b1        44 seconds ago      20.6MB
```

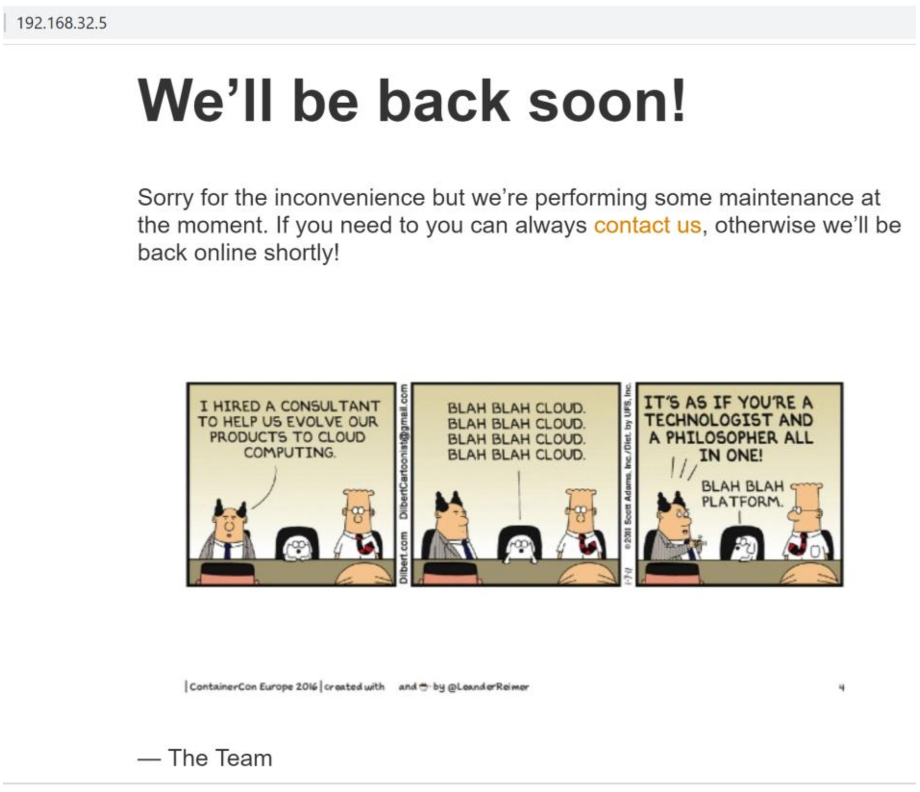
Run the container and specify the ports:

```
vagrant@devops1:~/src$ sudo docker run -d -p 80:80 meetup-app
91f42dc899dff09b4702227f5fc05409d0c79bd8905cf21b082e9702e2b806f7

vagrant@devops1:~/src$ sudo docker ps
CONTAINER ID  IMAGE      COMMAND       CREATED  STATUS    PORTS                NAMES
91f42dc899df  meetup-app  "nginx -g …"  10 secs  Up 8 secs 0.0.0.0:80->80/tcp  determined_germain
```

Let's check if it worked:

```
vagrant@devops1:~/src$ curl localhost:80
```

You should see the index.html page printed out to the screen.
We can also visit the webpage using the VM IP address from your web browser:



192.168.32.5

We'll be back soon!

Sorry for the inconvenience but we're performing some maintenance at the moment. If you need to you can always contact us, otherwise we'll be back online shortly!

I HIRED A CONSULTANT TO HELP US EVOLVE OUR PRODUCTS TO CLOUD COMPUTING.
BLAH BLAH CLOUD. BLAH BLAH CLOUD. BLAH BLAH CLOUD. BLAH BLAH CLOUD.
IT'S AS IF YOU'RE A TECHNOLOGIST AND A PHILOSOPHER ALL IN ONE!
BLAH BLAH PLATFORM.
|ContainerCon Europe 2016|created with and by @LeanderReimer

— The Team

- You've created your first container to host a static website.

Pushing containers to docker hub:

Now you've created a new container and tested it, let's push it up to docker hub so we can pull it down and use it later on.

Steps to complete:

Create a Docker-hub account or login to your existing account here: https://hub.docker.com

- On your VM, login to docker hub via the command line:

```
vagrant@devops1:~/src$ sudo docker login --username=<your-userame>
Login Succeeded
```

- Check the docker image you want to upload: `$ sudo docker images`

```
REPOSITORY       TAG          IMAGE ID         CREATED              SIZE
meetup-app        latest       1e743fba89b1      15 minutes ago       20.6MB
```

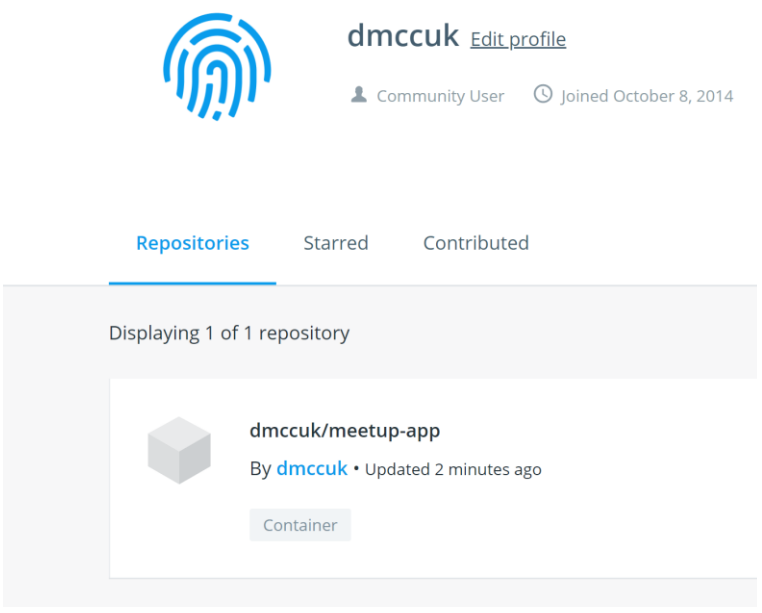
- Tag your image:

```
vagrant@devops1:~/src$ sudo docker tag 1e743fba89b1 dmccuk/meetup-app:first
```

- Docker push username/repository.

```
vagrant@devops1:~/src$ sudo docker push dmccuk/meetup-app:first
The push refers to repository [docker.io/dmccuk/meetup-app]
9acad8977816: Pushed
fbe0fc9bcf95: Mounted from dmccuk/meetup-app
f1b5933fe4b5: Mounted from dmccuk/meetup-app
first: digest:
sha256:2393c5afcec5a2d284c37b26f621d88be649fb98e049920dc70237adfe7143ca size:
946
```

Once the upload has finished, check your docker hub repository and confirm the container has been uploaded.


dmccuk Edit profile

Community User   Joined October 8, 2014

Repositories   Starred   Contributed

Displaying 1 of 1 repository

dmccuk/meetup-app

By dmccuk • Updated 2 minutes ago

Container

Docker Summary:

- Created our first container.
- We've started it and can see it running.
- Uploaded our container to Docker-Hub.
- It's now available to the world.

Kubernetes: What is it?

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services that facilitates both declarative configuration and automation.

Why do I need it?

Containers are a good way to bundle and run your applications. In a production environment, you need to manage the containers that run the applications and ensure that there is no downtime. For example, if a container goes down, another container needs to start. Wouldn't it be easier if this behaviour was handled by a system?

That's how Kubernetes comes to the rescue! Kubernetes provides you with a framework to run distributed systems resiliently. It takes care of scaling and failover for your application, provides deployment patterns, and more.

Source: https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

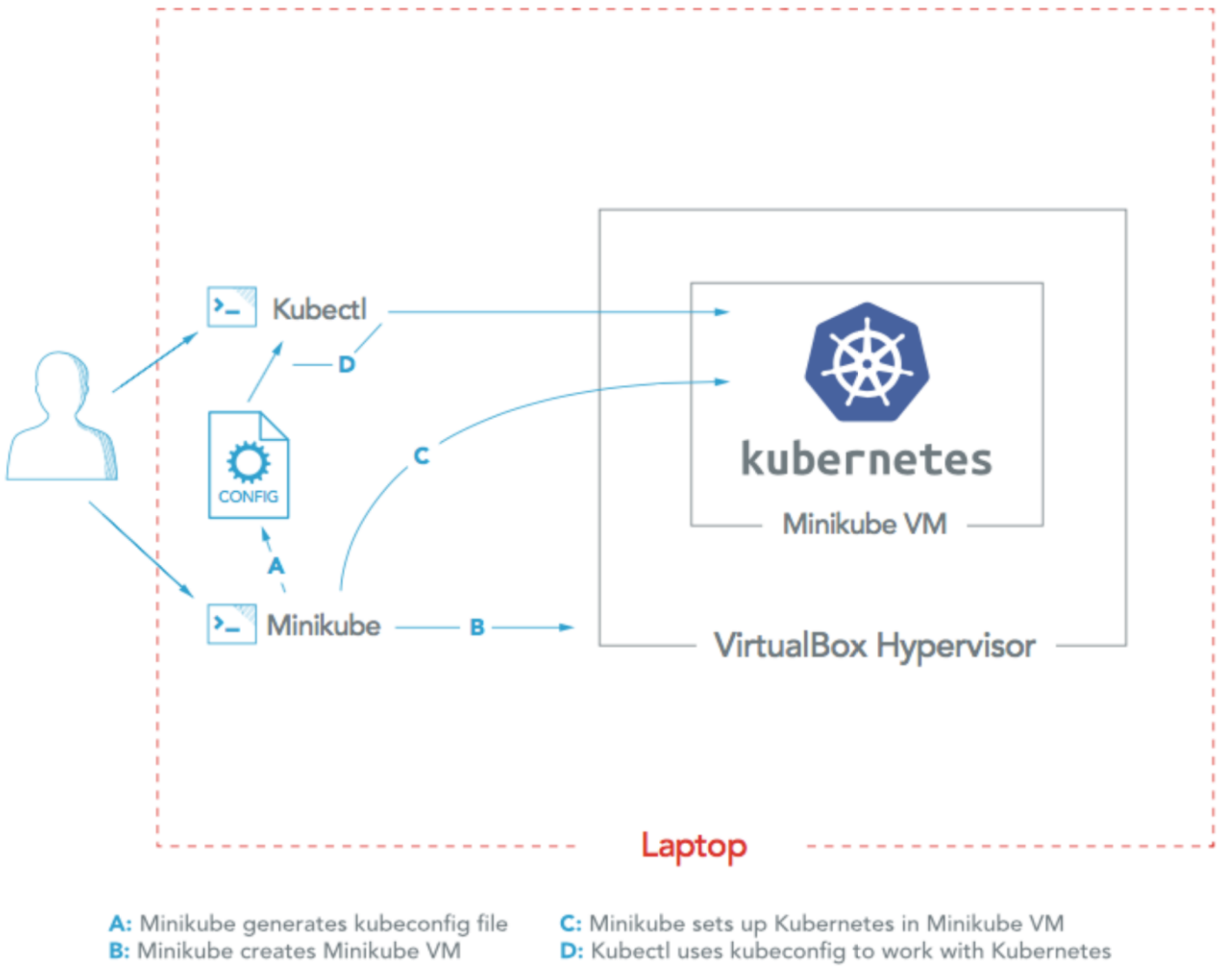
Introduction to minikube

Minikube is a tool that makes it easy to run Kubernetes locally. Minikube runs a single-node Kubernetes cluster inside a Virtual Machine (VM) on your laptop for users looking to try out Kubernetes or develop with it day-to-day.

Start Minikube now (it takes a few minutes to start!)

In this example, we'll use our laptop along with Minikube, VirtualBox and Kubectl to create a single node Kubernetes cluster:
(Source picture:Platform9: https://platform9.com/docs/install-kubernetes-the-ultimate-guide/)

Overview:


Kubectl
D
CONFIG
A
C
Minikube
B
kubernetes
Minikube VM
VirtualBox Hypervisor
Laptop
A: Minikube generates kubeconfig file
B: Minikube creates Minikube VM
C: Minikube sets up Kubernetes in Minikube VM
D: Kubectl uses kubeconfig to work with Kubernetes

Minikube install:

On windows, MAC OS or Linux, download and install minikube from the following link:
https://github.com/kubernetes/minikube

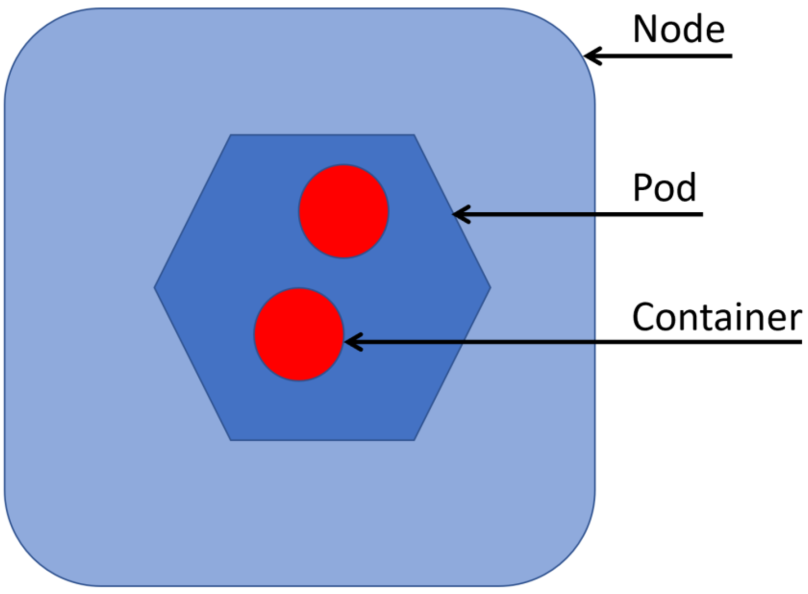
Follow the install procedure first and once complete, continue with these steps.

```
C:\Program Files\Kubernetes\Minikube>minikube start
* minikube v1.2.0 on windows (amd64)
* Tip: Use 'minikube start -p <name>' to create a new cluster, or 'minikube delete' to
delete this one.
* Re-using the currently running virtualbox VM for "minikube" ...
* Waiting for SSH access ...
* Configuring environment for Kubernetes v1.15.0 on Docker 18.09.6
* Relaunching Kubernetes v1.15.0 using kubeadm ...
* Verifying: apiserver proxy etcd scheduler controller dns
* Done! kubectl is now configured to use "minikube"
* For best results, install kubectl: https://kubernetes.io/docs/tasks/tools/install-
kubectl/

C:\Program Files\Kubernetes\Minikube>
```

What is a Pod?

A Kubernetes pod is a group of containers that are deployed together on the same host. If you frequently deploy single containers, you can generally replace the word "pod" with "container" and accurately understand the concept.

Pods operate at one level higher than individual containers because it's very common to have a group of containers work together to produce an artefact or process a set of work.



Kubectl install:

On windows, MAC OS or Linux, download and install kubectl from the following link:
https://kubernetes.io/docs/tasks/tools/install-kubectl/

Find your OS and follow the install process. I moved the kubectl.exe to the minikube directory (or you can add it to your $PATH).

Kubernetes commands

Run the following command:

```
C:\Program Files\Kubernetes\Minikube> kubectl version
```

The kubectl command creates a deployment and deployments create our pods and keep them up and running.

```
C:\Program Files\Kubernetes\Minikube> kubectl run meetup-app --image=dmccuk/meetup-
app:first --port=80
deployment.apps/maint-app created
```

Now let's check our running pod:

```
C:\Program Files\Kubernetes\Minikube> kubectl get pods
      NAME                           READY   STATUS        RESTARTS   AGE
      meetup-app-9f4989dff-98gw4      0/1     Running        0          55s

C:\Program Files\Kubernetes\Minikube> kubectl get deployment
      NAME         READY   UP-TO-DATE   AVAILABLE   AGE
      meetup-app    1/1     1             1           48s
```

Lets check out the deployment configuration by checking the yaml file:

```
C:\Program Files\Kubernetes\Minikube> kubectl get deployment --export -o yaml
Flag --export has been deprecated, This flag is deprecated and will be removed in future.
      apiVersion: v1
      items:
      - apiVersion: extensions/v1beta1
        kind: Deployment
        metadata:
          annotations:
            deployment.kubernetes.io/revision: "1"
          creationTimestamp: "2019-08-04T07:58:08Z"
          generation: 1
          labels:
            run: meetup-app
          name: meetup-app
      <results omitted>
```

Now it's running we can expose it outside of Kubernetes:

```
C:\Program Files\Kubernetes\Minikube> kubectl expose deployment meetup-app --type=NodePort
service/meetup-app exposed
```

Other options are available to expose your deployment. These include Port-forwarding, NodePort & Load-balancer.

Once we expose out app, we can check the service. There is our exposed IP address:

```
C:\Program Files\Kubernetes\Minikube> kubectl get service
NAME         TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP        26d
meetup-app   NodePort    10.98.35.163   <none>        80:31450/TCP   35s
```

Now get the URL and check it works:

```
C:\Program Files\Kubernetes\Minikube> minikube service meetup-app --url
```

```
        http://192.168.99.100:31000
```

Let’s scale out application:

```
        C:\Program Files\Kubernetes\Minikube> kubectl scale deployment meetup-app --replicas=3
        deployment.extensions/meetup-app scaled
```

And to prove it:

```
        C:\Program Files\Kubernetes\Minikube> kubectl get po
        NAME                              READY    STATUS     RESTARTS    AGE
        meetup-app-7f794c5cf7-bdtdv       1/1      Running    0           8s
        meetup-app-7f794c5cf7-dnv9w       1/1      Running    0           8s
        meetup-app-7f794c5cf7-khzs4       1/1      Running    0           6m30s
```

We can also get the individual end points:

```
        C:\Program Files\Kubernetes\Minikube> kubectl get ep meetup-app
        NAME         ENDPOINTS                                          AGE
        meetup-app    172.17.0.5:80,172.17.0.6:80,172.17.0.7:80         7m6s

        C:\Program Files\Kubernetes\Minikube> kubectl get po -o wide
NAME                          READY  STATUS     RESTARTS   AGE     IP          NODE       NOMINATED NODE   READINESS
GATES
meetup-app-7f794c5cf7-bdtdv    1/1    Running    0          2m46s  172.17.0.6  minikube  <none>           <none>
meetup -app-7f794c5cf7-dnv9w   1/1    Running    0          2m46s  172.17.0.7  minikube  <none>           <none>
meetup -app-7f794c5cf7-khzs4   1/1    Running    0          9m8s   172.17.0.5  minikube  <none>           <none>
```

Testing: If we delete 2 of the pods, what’s going to happen?

```
        C:\Program Files\Kubernetes\Minikube> kubectl delete po meetup-app-7f794c5cf7-5h4gg maint-
        app-7f794c5cf7-lxsxg
        pod "meetup-app-7f794c5cf7-5h4gg" deleted
        pod "meetup-app-7f794c5cf7-lxsxg" deleted

        C:\Program Files\Kubernetes\Minikube> kubectl get po
        NAME                             READY    STATUS     RESTARTS    AGE
        meetup-app-7f794c5cf7-ghzqq       1/1      Running    0           65s
        meetup-app-7f794c5cf7-j64zt       1/1      Running    0           18s
        meetup-app-7f794c5cf7-rkvws       1/1      Running    0           18s
```

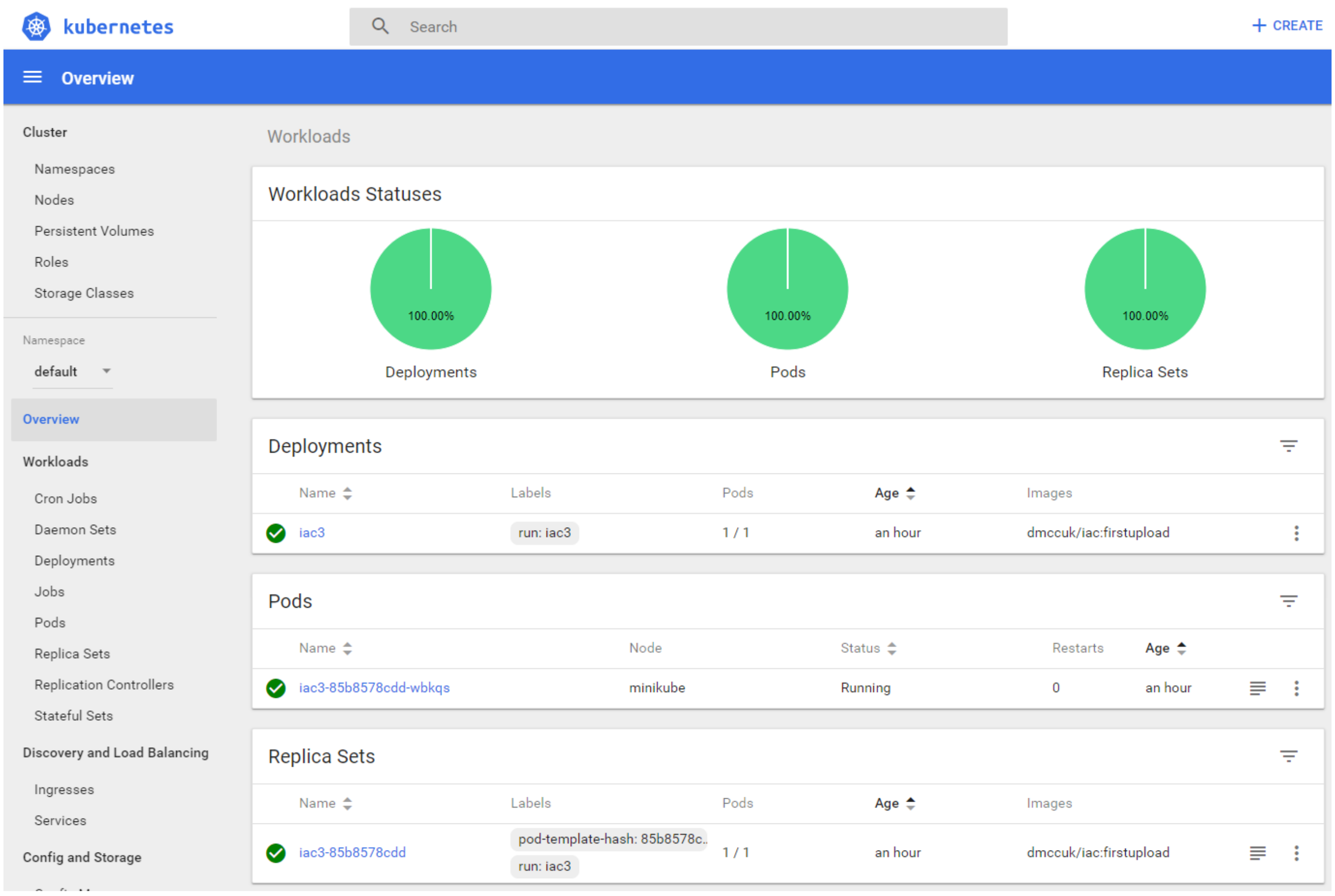
Now let go to the service URL via minikube:

```
        C:\Program Files\Kubernetes\Minikube> minikube service meetup-app --url
        http://192.168.99.100:32294

        C:\Program Files\Kubernetes\Minikube> minikube service meetup-app
        * Opening kubernetes service default/meetup-app in default browser...


        C:\Program Files\Kubernetes\Minikube> kubectl get svc
               NAME          TYPE         CLUSTER-IP     EXTERNAL-IP    PORT(S)         AGE
               kubernetes    ClusterIP    10.96.0.1      <none>         443/TCP         20d
               meetup-app    NodePort     10.100.0.46    <none>         80:32294/TCP    109s

        C:\Program Files\Kubernetes\Minikube> kubectl describe svc meetup-app
        Name:                     maint-app
        Namespace:                default
        Labels:                   run=meetup-app
        Annotations:              <none>
        Selector:                 run=meetup-app
        Type:                     NodePort
        IP:                       10.100.0.46
        Port:                     <unset>  80/TCP
        TargetPort:               80/TCP
        NodePort:                 <unset>  32294/TCP
        Endpoints:                172.17.0.5:80
        Session Affinity:         None
        External Traffic Policy:  Cluster
        Events:                   <none>
```

Let checkout the dashboard:

```
C:\Program Files\Kubernetes\Minikube> minikube dashboard
* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
* Opening http://127.0.0.1:60417/api/v1/namespaces/kube-system/services/http:kubernetes-
dashboard:/proxy/ in your default browser...
```

A browser will open and you will see the cluster and information displayed on the webpage.



Deployments

Name	Labels	Pods	Age	Images
iac3	run: iac3	1 / 1	an hour	dmccuk/iac:firstupload

Pods

Name	Node	Status	Restarts	Age
iac3-85b8578cdd-wbkqs	minikube	Running	0	an hour

Replica Sets

Name	Labels	Pods	Age	Images
iac3-85b8578cdd	pod-template-hash: 85b8578c.. run: iac3	1 / 1	an hour	dmccuk/iac:firstupload

Either close the browser or press “Ctrl + c” to close it from the command line.

What have you accomplished?

- Built a docker container for a static website.
- Pushed your container to the Docker hub
- Installed and setup minikube and kubectl on you Laptop
- Started a Kubernetes cluster an ran the container image
- Exposed the container deployment and port
- Used minikube to start the container as a Kubernetes service adding replicas
- Finally started up the Kubernetes dashboard to display information about our cluster and pod.

Delete deployment and service:

Now that we've used Kubernetes to run out container, we can delete the deployment and service.

First, let's get all the information from Kubernetes that we need to delete it:

```
C:\Program Files\Kubernetes\Minikube>kubectl get all
NAME                                 READY   STATUS    RESTARTS   AGE
pod/meetup-app-5c4bc454f8-2cv2j      1/1     Running   0          4m9s
pod/meetup-app-5c4bc454f8-d5xgj      1/1     Running   0          6m36s
pod/meetup-app-5c4bc454f8-xm67x      1/1     Running   0          4m9s


NAME                   TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
service/kubernetes     ClusterIP   10.96.0.1       <none>        443/TCP        41d
service/meetup-app     NodePort    10.108.228.66   <none>        80:31000/TCP   9m31s


NAME                          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/meetup-app    3/3     3            3           13m

NAME                                     DESIRED   CURRENT   READY   AGE
replicaset.apps/meetup-app-5c4bc454f8    3         3         3       13m
```

Now, run this command:

```
C:\Program Files\Kubernetes\Minikube>kubectl delete deploy/meetup-app svc/meetup-app
deployment.extensions "meetup-app" deleted
service "meetup-app" deleted
```

All that's left is the minikube cluster.

```
C:\Program Files\Kubernetes\Minikube>kubectl get all

NAME                   TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes     ClusterIP   10.96.0.1    <none>        443/TCP   20h
```

Lets stop Minikube:
m

```
C:\Program Files\Kubernetes\Minikube>minikube stop
* Stopping "minikube" in virtualbox ...
* "minikube" stopped.
```

Stop and remove the container

```
sudo docker container ls -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
PORTS               NAMES
340d593aa625        61b9b2dd753e        "nginx -g 'daemon of…"   10 minutes ago      Up 10
minutes       0.0.0.0:80->80/tcp   musing_cray

sudo docker stop 340d593aa625
340d593aa625

sudo docker container rm 340d593aa625
340d593aa625

sudo docker container ls -a
CONTAINER ID    IMAGE      COMMAND        CREATED            STATUS             PORTS
NAMES

sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
meetup-app          latest              d7fc29e4cd71        2 minutes ago       21.9MB
dmccuk/meetup-app   first               61b9b2dd753e        12 minutes ago      21.9MB
nginx               alpine              48c8a7c47625        2 weeks ago         21.8MB

sudo docker system prune -a
WARNING! This will remove:
        - all stopped containers
        - all networks not used by at least one container
        - all images without at least one container associated to them
        - all build cache
Are you sure you want to continue? [y/N] y
Deleted Images:
untagged: nginx:alpine
untagged: nginx@sha256:9e81b8f9cef5a095f892183688798a5b2c368663276aa0f2be4b1cd283ace53d
untagged: dmccuk/meetup-app:first
untagged: dmccuk/meetup-
app@sha256:f57b48e71553e13e7933afaa9de9f96b45c3b3a4159a050682c35118c134214c
deleted: sha256:61b9b2dd753e18ea89487eba30efea72c0ca021c54afc984c23e7f3132bf7696
deleted: sha256:7d12e4a4b4571e9f8a80076a543e5b7f96c9564b8c53e4a96b8d6c7dadd7b908
deleted: sha256:904a6fbf3c85bc59d5ab586ebe8ddf0c36ca7912967f8941e1298608f0f76aa9
untagged: meetup-app:latest
deleted: sha256:d7fc29e4cd7128a051f5e3582cb46b053b7b36b986369332f74f4d609706e6d7
deleted: sha256:1bc62ddc6b675ae7f2701917cb8409ca07900d64693ddbbf3618931d631442c5
deleted: sha256:d23509eb0c10ee3c582f1f9a4eff62d80ae1e791b5b44a34290f3cd09c4a73d3
deleted: sha256:48c8a7c476256c69882b00a91cc225c54bd29c963fb4f8ce2581c7286a52fadc
deleted: sha256:9bcf685af3e7ce8e0df0361544f12cb2c015b703927f5f0f9a8fd3a15a4bd59f
deleted: sha256:531743b7098cb2aaf615641007a129173f63ed86ca32fe7b5a246a1c47286028

Total reclaimed space: 21.92MB
```

Script to build the Docker image:

```
cd
mkdir src
cd src
cat > index.html << EOF
<!doctype html>
<title>LondonIAC meetup - Site Maintenance</title>
<style>
  body { text-align: center; padding: 150px; }
  h1 { font-size: 50px; }
  body { font: 20px Helvetica, sans-serif; color: #333; }
  article { display: block; text-align: left; width: 650px; margin: 0 auto; }
  a { color: #dc8100; text-decoration: none; }
  a:hover { color: #333; text-decoration: none; }
</style>
```

```
<article>
    <h1>We&rsquo;ll be back soon!</h1>
    <div>
        <p>Sorry for the inconvenience but we&rsquo;re performing some maintenance at the
moment. If you need to you can always <a href="mailto:#">contact us</a>, otherwise we&rsquo;ll
be back online shortly!</p>
    <img src="https://marcelorjava.files.wordpress.com/2014/04/dilbert.gif" alt="Dilbert">
        <p>&mdash; The Team</p>
    </div>
    <p>
</p>
</article>
EOF

cat >  Dockerfile << EOF
FROM nginx:alpine
COPY . /usr/share/nginx/html
EXPOSE 80
EOF

echo "sudo docker build -t meetup-app ."
echo "sudo docker images"
echo "sudo docker run -d -p 80:80 meetup-app"
echo "sudo docker ps"
echo "sudo docker login --username=dmccuk"
echo "sudo docker images"
echo "sudo docker tag TAG dmccuk/meetup-app:TAG"
echo "sudo docker push dmccuk/meetup-app:TAG"
```