

A cluster of overlapping triangles in red, purple, and teal colors located in the top-left corner of the slide.

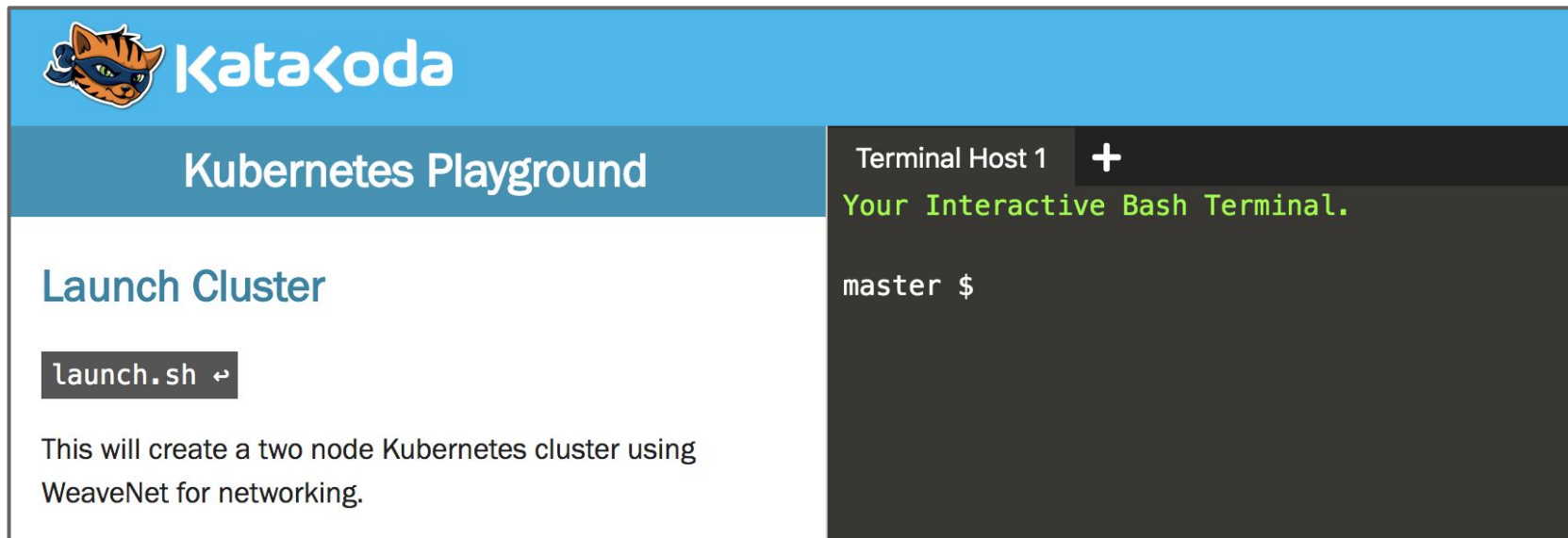
Demonstration

A series of overlapping triangles in various shades of purple and blue, located in the bottom-left corner of the slide.

Overview

1. Kubernetes Environment
2. Create Docker Hub account
3. Copy Workshop Files
4. Build and Publish a Docker image
5. Create a Kubernetes **Deployment**

Kubernetes Environment



The image shows the KataCoda Kubernetes Playground interface. It features a blue header with the KataCoda logo (a cat wearing a mask) and the text "KataCoda". Below the header, there is a section titled "Kubernetes Playground". On the left side of this section, there is a "Launch Cluster" button with a "launch.sh" icon. Below the button, there is a text description: "This will create a two node Kubernetes cluster using WeaveNet for networking." On the right side of the interface, there is a terminal window titled "Terminal Host 1" with a "+" icon. The terminal displays the text "Your Interactive Bash Terminal." and "master \$".

KataCoda

Kubernetes Playground

Launch Cluster

`launch.sh ↵`

This will create a two node Kubernetes cluster using WeaveNet for networking.

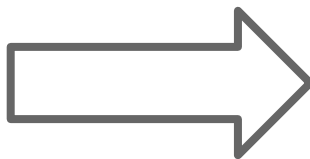
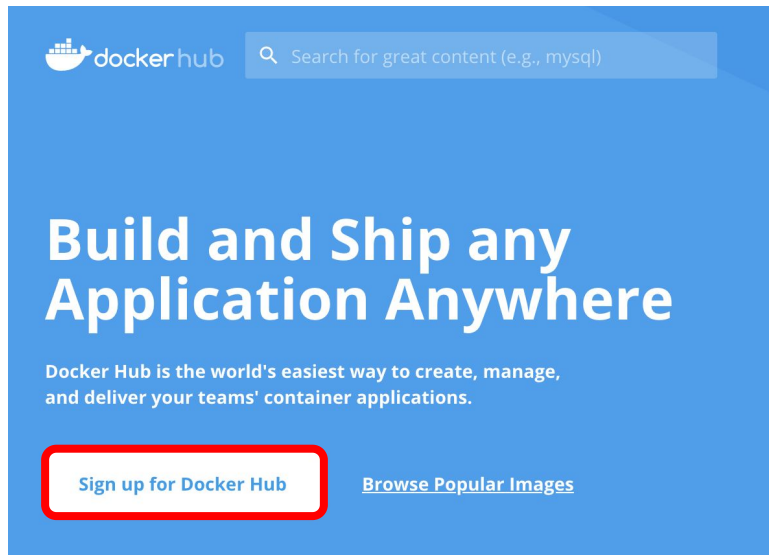
Terminal Host 1 +

Your Interactive Bash Terminal.

master \$

katacoda.com/courses/kubernetes/playground

Create a Docker Hub account

A screenshot of the "Create a Docker ID" form on Docker Hub. The form is white with a blue border. At the top is the Docker logo. Below it, the heading "Create a Docker ID." is followed by the text "Already have an account? [Sign In](#)". The form contains three input fields: "Docker ID", "Email", and "Password". Below these fields are three checkboxes: "I agree to Docker's [Terms of Service](#).", "I agree to Docker's [Privacy Policy](#) and [Data Processing Terms](#).", and "Send me occasional product updates and announcements.". At the bottom left is a checkbox for "I'm not a robot" next to a reCAPTCHA logo. A large blue "Sign Up" button is at the bottom right.

hub.docker.com

Get the Workshop Files

```
Terminal Host 1 +
master $ curl -O https://raw.githubusercontent.com/appian/hackathons/master/workshops/kubernetes/server.py
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Dload  % Upload   Total   Spent    Left     Speed
100  535  100  535    0     0  1280      0 --:--:-- --:--:-- --:--:-- 1279
master $ ls
go server.py
master $ curl -O https://raw.githubusercontent.com/appian/hackathons/master/workshops/kubernetes/Dockerfile
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Dload  % Upload   Total   Spent    Left     Speed
100  134  100  134    0     0   200      0 --:--:-- --:--:-- --:--:--  200
master $ curl -O https://raw.githubusercontent.com/appian/hackathons/master/workshops/kubernetes/deployment.yaml
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Dload  % Upload   Total   Spent    Left     Speed
100  414  100  414    0     0   557      0 --:--:-- --:--:-- --:--:--  567
```

[server.py](#), [Dockerfile](#), and [deployment.yaml](#)

Build and publish a Docker image

server.py

```
from http.server import HTTPServer, BaseHTTPRequestHandler

class RequestHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        self.send_response(200)
        self.send_header('Content-type', 'text/html')
        self.end_headers()
        self.wfile.write("<html><h1>Kubernetes (and Appian) is  
pretty cool.</h1></html>".encode())

if __name__ == '__main__':
    server_address = ('', 8080)
    print('Listening on {}'.format(server_address))
    server = HTTPServer(server_address, RequestHandler)
    server.serve_forever()
```

Build and publish a Docker image

Dockerfile

```
FROM python:3.6-slim

COPY ./server.py /server.py

RUN adduser --uid 1000 --system user
USER 1000

ENTRYPOINT ["python", "server.py"]
```

Build and publish a Docker image

DON'T FORGET THE PERIOD

```
docker build .  
  
docker login --username <username>  
docker tag <SHA>  
<username>/basic-server:<TAG>  
docker push <username>/basic-server:<TAG>
```


Create a Deployment in the cluster

deployment.yaml (pt. 1)

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: <name>
  labels:
    app: <name>
```

deployment.yaml (pt. 2)

```
spec:
  replicas: 1
  selector:
    matchLabels:
      app: <name>
  template:
    metadata:
      labels:
        app: <name>
    spec:
      containers:
        - name: server
          image: <username>/basic-server:<tag>
          ports:
            - containerPort: 8080
```

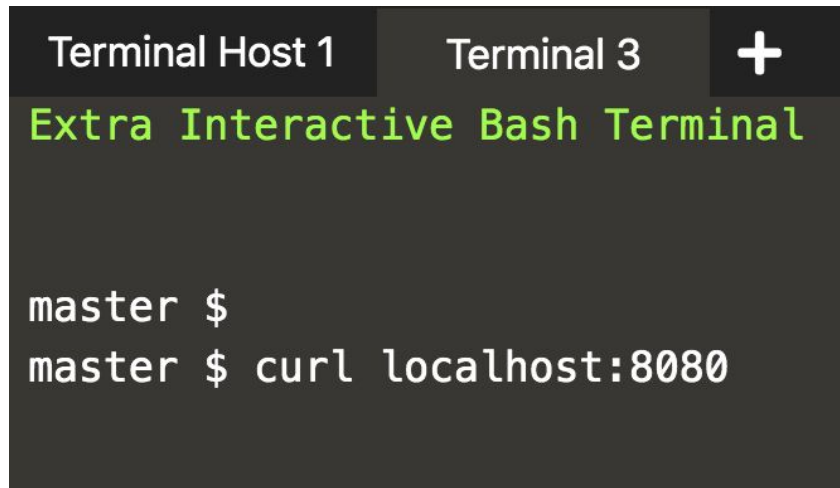
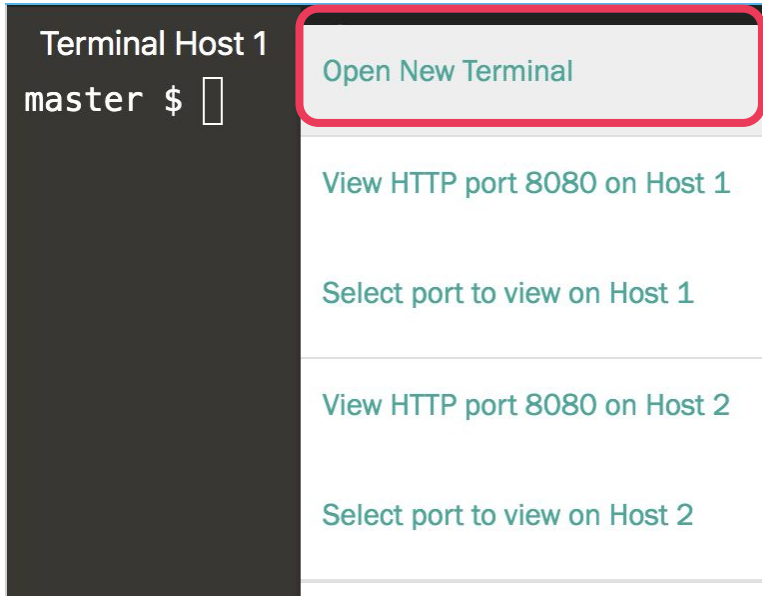
Create a Deployment in the cluster

```
kubectl apply -f deployment.yaml
```

```
kubectl get pods
```

```
kubectl port-forward <pod> 8080
```

Create a Deployment in the cluster





Thank you!

Questions?

Appian