```
$ curl XXX.XXX.XXX.XXX
Hello World! ...
```

1. You can also verify that the limits specified in the `service.yaml` file are in effect with `kubectl describe pod XXX` .

Change the service

1. You've discovered your service is on Hacker News, and you want to bump up the memory and change your greeting. Edit the `service.yaml` file and change the memory to 0.5G. ProTip: if you don't specify a limit, Kubernetes will default to unlimited ... which will enable an errant service to take down your entire cluster.

   So let's change some source code and redeploy:

   ```
   sed -i -e 's/Hello World!/Hello Hacker News!!!/' hello-forge/app.py
   forge deploy
   ```

2. Now you can curl and see the new message (Kubernetes may take a few seconds to rollout the new image):

   ```
   $ curl XXX.XXX.XXX.XXX
   Hello Hacker News!!! ...
   ```

3. You can verify that the service does have more memory with `kubectl describe pods` , as above.

A network of services

1. So now we've seen we can easily build and deploy a single service, but microservices are truly useful when you can get a whole bunch of them to work together. Using Forge we can just as easily spin up a whole network of microservices:

   ```
   git clone https://github.com/datawire/hello-forge-network.git
   forge deploy
   ```

2. You can see Forge has built, pushed, and deployed the entire network of services.

   ```
   kubectl get services
   ```

Next steps

You've seen an example of how Forge can quickly build and deploy services to Kubernetes. Now, try setting up Forge on your own services.

Still have questions? Ask in our Gitter chatroom or file an issue on GitHub.