



Exercise 2: From Docker to Kubernetes

Each assignment is graded out of 5 points. The topic for this assignment is getting started with Kubernetes. This exercise will loosely follow this Kubernetes tutorial from the ComputeFest event: https://github.com/Harvard-IACS/2020-ComputeFest/tree/master/notebook_to_cloud/ml_deploy_demo/kubernetes

Question 1: Install kubectl (0.5 points)

Install the Kubernetes command-line tool, [kubectl](#), so that you can communicate with Kubernetes clusters. If you are running Ubuntu, the apt installation is cleaner than manually copying the kubectl executable onto your path.

Verify the installation by running the following two commands (on Linux/Mac; on Windows the equivalent utility to which is where).

```
$ which kubectl
```

```
$ kubectl version --client
```

Submit:

1. Screenshot with the output of the two above commands

Example Submission:

```
[4/6] Cmdr
(ac295) michael@Loki: /usr/local/bin 2020-02-10 21:05:03
$ which kubectl
/usr/bin/kubectl
(ac295) michael@Loki: /usr/local/bin 2020-02-10 21:05:07
$ kubectl version --client
Client Version: version.Info{Major:"1", Minor:"17", GitVersion:"v1.17.2", GitCommit:"59603c6e503c87169aea6106f57b9f242f6
4df89", GitTreeState:"clean", BuildDate:"2020-01-18T23:30:10Z", GoVersion:"go1.13.5", Compiler:"gc", Platform:"linux/amd
64"}
(ac295) michael@Loki: /usr/local/bin 2020-02-10 21:05:10
```



Question 2: Install Virtualbox (0.5 points)

Install virtualbox on your computer. This is a prerequisite for the minikube package we will use below. Instructions for the installation can be found at https://www.virtualbox.org/wiki/Linux_Downloads

Hint: If you are installing on Linux, there is one step that can be tricky. When you install the optional virtualbox-ext-pack package, there is a setup screen from dpkg that asks you to accept the license. It can appear that this screen is hanging. Navigate to the bottom with page down or the arrow keys; then select the “OK” button by hitting Tab, and accept the license by hitting Enter.

For my Linux (Ubuntu 19.10) the exact steps that worked were:

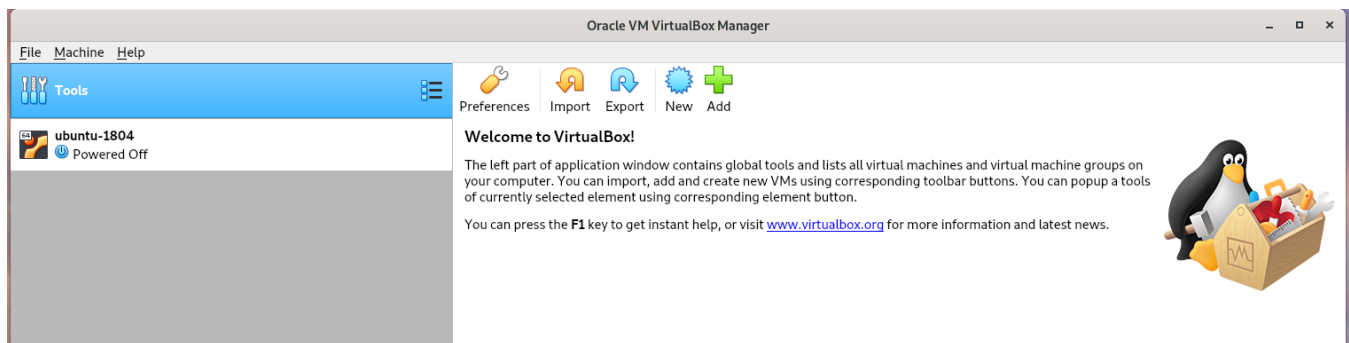
(1) enable the universe and multiverse repositories in the apt package manager; then

```
$ sudo apt install virtualbox virtualbox-ext-pack
```

Submit:

1. Screenshot showing VirtualBox successfully installed on your computer.

Example Submission:





Question 3: Install minikube (0.5 points)

Install minikube on your computer by following the instructions here:

<https://kubernetes.io/docs/tasks/tools/install-minikube/>

After you have installed minikube, start a cluster using the VirtualBox driver by issuing these two commands from the minikube installation instructions:

```
$ minikube config set vm-driver virtualbox
```

```
$ minikube start --vm-driver=virtualbox
```

Now check that everything is working with the status command like this:

```
$ minikube status
```

Hint: On my Linux machine, I needed to reboot after installing VirtualBox before this step worked. I think it's because the installation adds some modules to the Linux kernel, which only load on the next boot.

Submit:

1. Screenshot showing that minikube is running a cluster with the VirtualBox driver

Example Submission:

```
michael@Loki: ~/Software 2020-02-11 00:10:06
$ minikube start --vm-driver=virtualbox
🔍 minikube v1.7.2 on Ubuntu 19.10
👉 Using the virtualbox driver based on user configuration
🔧 Reconfiguring existing host ...
🔍 Starting existing virtualbox VM for "minikube" ...
🔍 Preparing Kubernetes v1.17.2 on Docker 19.03.5 ...
🔍 Downloading kubeadm v1.17.2
🔍 Downloading kubectl v1.17.2
🔍 Downloading kubelet v1.17.2
🔍 Launching Kubernetes ...
🔍 Enabling addons: default-storageclass, storage-provisioner
🔍 Done! kubectl is now configured to use "minikube"
michael@Loki: ~/Software 2020-02-11 00:11:48
$ minikube status
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```



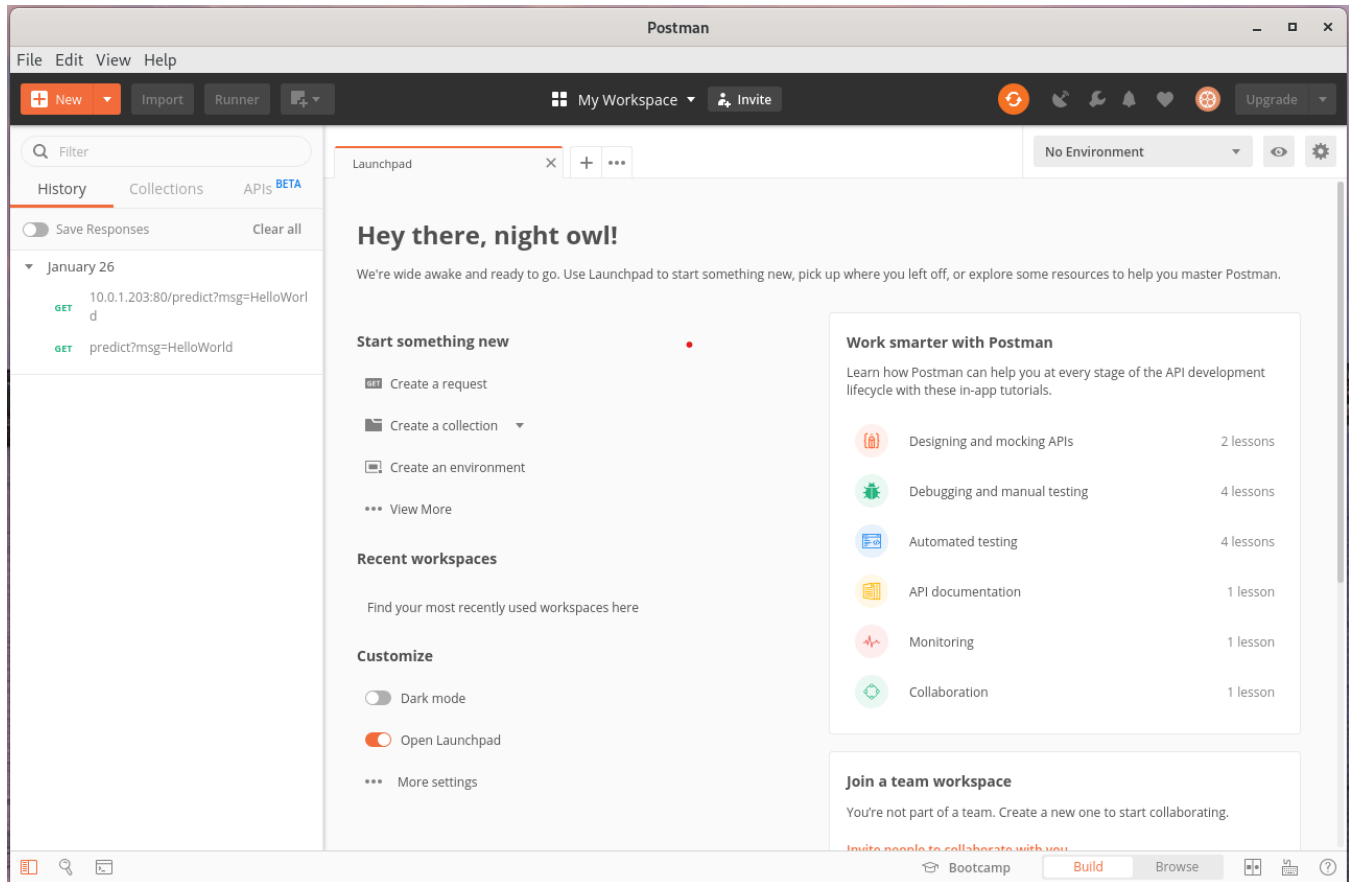
Question 4: Install Postman (0.5 points)

Install the Postman application by following the instructions here <https://www.postman.com/downloads/>

Submit:

1. Screenshot showing that Postman was successfully installed

Example Submission:





Question 5: Build the Sentiment Analysis App in Docker (1.5 points)

Follow the steps in the tutorial to build the sentiment classifier application in Docker on your local computer. Start by cloning the ComputeFest repository to a convenient location on your computer, e.g.
`$ git clone git@github.com:Harvard-IACS/2020-ComputeFest.git` ComputeFest-2020

Hints: This is straightforward, but you really need to follow all of the instructions carefully or it won't work. First make sure that the docker process is running in the command line session. You should type a command and receive output that looks like this:

```
michael@Thor: ~/IACS/ComputeFest-2020/notebook_to_cloud/ml_deploy_demo 2020-02-11 00:55:23
$ docker run -it --rm -p 5000:5000 memanuel/sentiment-analysis:latest /bin/bash ml_deploy_demo/run.sh
PWD /app
[2020-02-11 05:55:45 +0000] [1] [INFO] Starting gunicorn 20.0.4
[2020-02-11 05:55:45 +0000] [1] [INFO] Listening at: http://0.0.0.0:5000 (1)
[2020-02-11 05:55:45 +0000] [1] [INFO] Using worker: sync
[2020-02-11 05:55:45 +0000] [8] [INFO] Booting worker with pid: 8
```

The app has told us it's listening on port 5000 of the local host for HTTP requests.

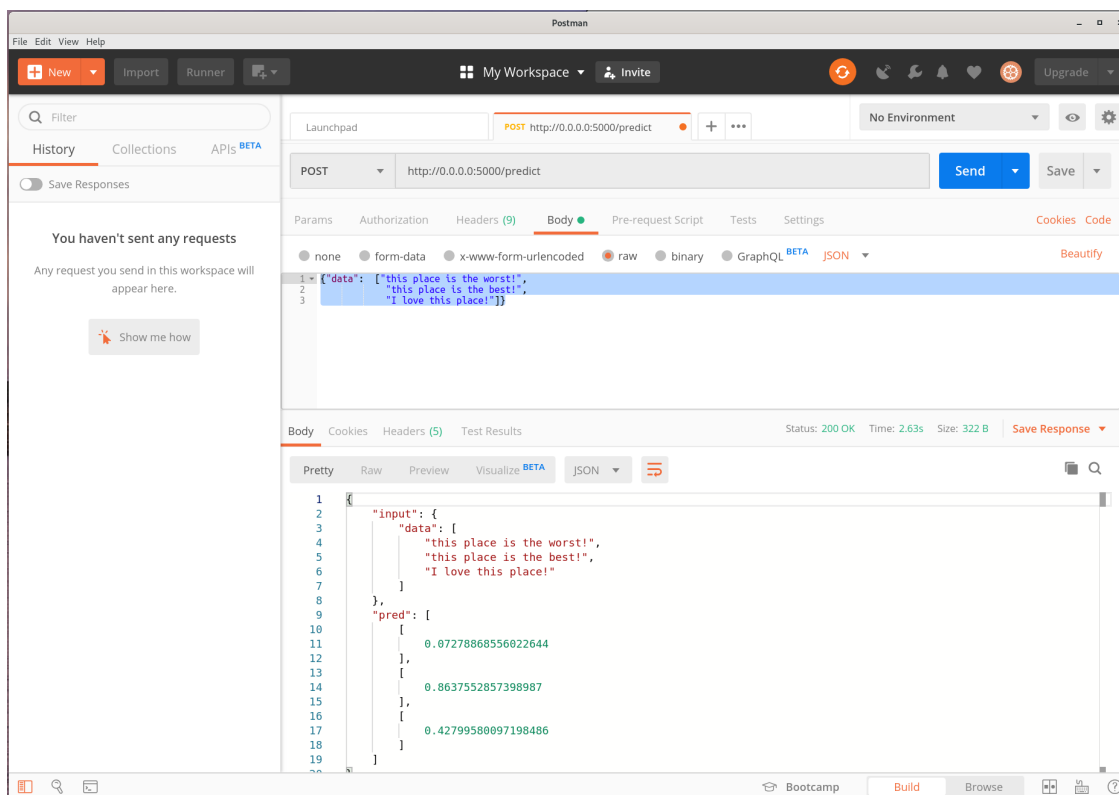
When you start up Postman, make sure that:

- 1) The request type is POST (the default is GET)
- 2) You select the "Body" tab on top of the menu
- 3) You choose an input type of "raw" under where you choose Body
- 4) You change the format to JSON on the right side of this menu

Submit:

1. Screenshot showing a successful call to your Docker sentiment analysis app

Example Submission:





Question 6: Deploy the Sentiment Analysis Service on Kubernetes (1.5 points)

Follow the steps of the tutorial to deploy your sentiment analysis app to a kubernetes cluster. The tutorial spins up a K8s cluster on AWS. That is somewhat resource intensive, and way overkill for this toy application. Minikube is more than enough for this task, and doesn't run the risk of incurring large bills if you forget to turn it off. We adapt this part of the exercise to use minikube instead of AWS.

Specific steps to take and example commands on Linux:

- 1) Create the kubernetes cluster using minikube if you haven't done so already, and verify that the master is running

```
$ minikube start
```

```
$ kubectl get nodes
```

```
michael@Loki: ~/IACS/ComputeFest-2020/notebook
$ kubectl get nodes
NAME        STATUS    ROLES    AGE   VERSION
minikube    Ready     master   10m   v1.17.2
```

- 2) Create the deployment and the service for the sentiment analysis app, following the ComputeFest example exactly. Wait a few seconds after creating the deployment until the pod shows a status of ready, then continue to the next step and create the service.

```
$ kubectl create -f sentiment_analysis_deployment.yaml
```

```
$ kubectl get pods
```

```
$ kubectl create -f sentiment_analysis_service.yaml
```

```
michael@Loki: ~/IACS/ComputeFest-2020/notebook_to_cloud/ml_deploy_demo/kubernetes 2020-02-11 01:41:05
$ kubectl create -f sentiment_analysis_service.yaml
service/sentiment-analysis-service created
michael@Loki: ~/IACS/ComputeFest-2020/notebook_to_cloud/ml_deploy_demo/kubernetes 2020-02-11 01:42:28
$ kubectl get svc
NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes                          ClusterIP           10.96.0.1       <none>           443/TCP          14m
sentiment-analysis-service          LoadBalancer       10.109.109.148  <pending>        5000:31734/TCP   10s
```

- 3) Expose the service as a URL in the minikube for testing, by entering this command:

```
$ minikube service sentiment-analysis-service
```

```
michael@Loki: ~/IACS/ComputeFest-2020/notebook_to_cloud/ml_deploy_demo/kubernetes 2020-02-11 01:54:14
$ minikube service sentiment-analysis-service
|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|
| default | sentiment-analysis-service | | http://192.168.99.101:31734 |
|-----|-----|-----|
[ ] Opening service default/sentiment-analysis-service in default browser...
```

- 4) Test that the app works on Postman, by submitting your request to the URL of the service running on minikube.



Submit:

A screenshot of the successful call to Postman

Partial credit is available if you can't get the call to Postman. Show screenshots showing how far you got and explain where things went wrong.

Example Submission:

Please note that the URL is 192.168.99.101:31734; this is a private IP and port that belongs to the minikube cluster.





Body Cookies Headers (5) Test Results

Pretty

Raw

Preview

Visualize **BETA**

JSON ▼

```
1  {
2    "input": {
3      "data": [
4        "this place is the worst!",
5        "this place is the best!",
6        "I love this place!"
7      ]
8    },
9    "pred": [
10     [
11       0.07278868556022644
12     ],
13     [
14       0.8637552857398987
15     ],
16     [
17       0.42799580097198486
18     ]
19   ]
}
```