

Docker image creation:

<https://github.com/aramaraj/docks-python-image>

Minikube Start

minikube start

Get Kubernetes cluster info :

1. `kubectl cluster-info`

Kubernetes master is running at https://192.168.99.100:8443

KubeDNS is running at https://192.168.99.100:8443/api/v1/proxy/namespaces/kube-system/services/kube-dns

kubernetes-dashboard is running at https://192.168.99.100:8443/api/v1/proxy/namespaces/kube-system/services/kubernetes-dashboard

2. The above address would give unauthorized so you need to proxy with the

`kubectl proxy --address="0.0.0.0" --port=9090`

1. in Browser access the port using local host
2. <http://127.0.0.1:9090/api/v1/proxy/namespaces/kube-system/services/kubernetes-dashboard/#/pod?namespace=default>

Create the Deployment:

`kubectl run docker-python-app --image=aramaraj/docker-python-app --port=8080`

kubectl get deployments

```
m-C02S23PLG8WM:docker-python-app aramar1$ kubectl get deployments
NAME                DESIRED    CURRENT    UP-TO-DATE    AVAILABLE    AGE
docker-python-app    1          1          1             1            3m
```

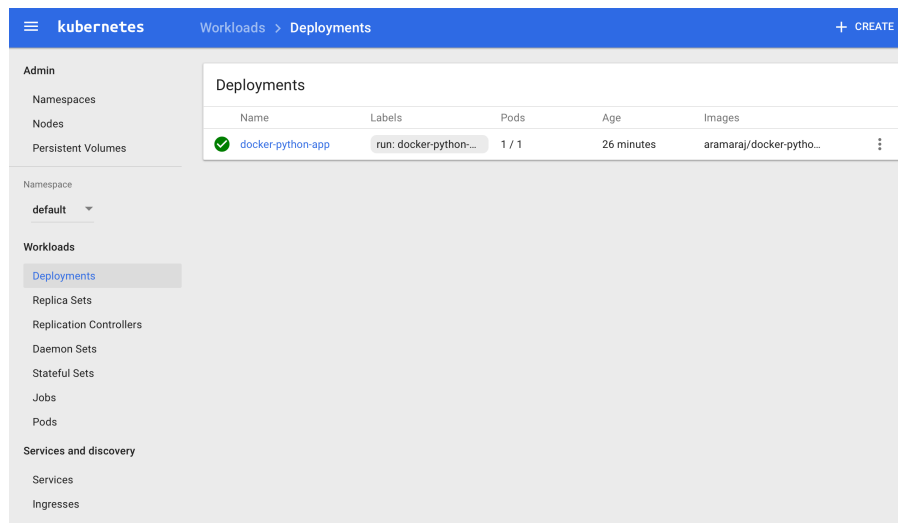
check the status of the pod created:

```
m-C02S23PLG8WM:docker-python-app aramar1$ kubectl get pods -o wide
NAME                                READY    STATUS    RESTARTS    AGE    IP
NODE
docker-python-app-3657222991-
t5kb9    1/1      Running    0           3m     172.17.0.4    minikube
```

Check the Events: (event log)

`kubectl get events`

Check the Deployment in Dash board:



kubernetes Workloads > Deployments + CREATE					
Admin Namespaces Nodes Persistent Volumes Namespace default Workloads Deployments Replica Sets Replication Controllers Daemon Sets Stateful Sets Jobs Pods Services and discovery Services Ingresses	Deployments				
	Name	Labels	Pods	Age	Images
	✓ docker-python-app	run: docker-python-...	1 / 1	26 minutes	aramaraj/docker-pytho...

Check the configuration:

kubectl config view

```
m-C02S23PLG8wM:docker-python-app aramar1$ kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /Users/aramar1/.minikube/ca.crt
    server: https://192.168.99.100:8443
    name: minikube
contexts:
- context:
    cluster: minikube
    user: minikube
    name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
  user:
    client-certificate: /Users/aramar1/.minikube/apiserver.crt
    client-key: /Users/aramar1/.minikube/apiserver.key
```

Check if the application runs on the POD:

```
m-C02S23PLG8WM:docker-python-app aramar1$ minikube ssh
```

```
$ curl 172.17.0.4:5000/track?name=ashok
```

Location of the Delivery truck number ashok is 315-317 N 10th St, San Jose, CA 95112, USA and Map URL is <http://maps.google.com/?q=37.345622600,-121.884722400>

The screenshot shows the Kubernetes dashboard's 'Logs' section. The left sidebar contains navigation links for Admin (Namespaces, Nodes, Persistent Volumes), Namespace (default), Workloads (Deployments, Replica Sets, Replication Controllers, Daemon Sets, Stateful Sets, Jobs, Pods), Services and discovery (Services, Ingresses), Storage (Persistent Volume Claims), and Config (Secrets). The main panel displays 'Logs from docker-python-app' in the 'docker-python-app-3657222991-t5kb9' pod. The logs show a series of timestamps followed by JSON data representing location tracking information. The final log entry is a GET request to the /track?name=ashok endpoint, returning an HTTP 1.1 status.

```
2017-04-07T05:28:28.794457570Z      "lng": -124.482003
2017-04-07T05:28:28.794459757Z      }
2017-04-07T05:28:28.794461867Z      },
2017-04-07T05:28:28.794463981Z      "location": {
2017-04-07T05:28:28.794466192Z        "lat": 36.778261,
2017-04-07T05:28:28.794468409Z        "lng": -119.4179324
2017-04-07T05:28:28.794470615Z      },
2017-04-07T05:28:28.794472713Z      "location_type": "APPROXIMATE",
2017-04-07T05:28:28.794475020Z      "viewport": {
2017-04-07T05:28:28.794477227Z        "northeast": {
2017-04-07T05:28:28.794479435Z          "lat": 42.009378,
2017-04-07T05:28:28.794481670Z          "lng": -114.131211
2017-04-07T05:28:28.794483940Z        },
2017-04-07T05:28:28.794486052Z        "southwest": {
2017-04-07T05:28:28.794488271Z          "lat": 32.5342852,
2017-04-07T05:28:28.794490526Z          "lng": -124.4151821
2017-04-07T05:28:28.794492778Z        }
2017-04-07T05:28:28.794494901Z      }
2017-04-07T05:28:28.794497052Z    },
2017-04-07T05:28:28.794499185Z    "place_id": "ChIJPV4oX_65j4ARVW8IJ6IJUys",
2017-04-07T05:28:28.794501524Z    "types": [
2017-04-07T05:28:28.794503730Z      "administrative_area_level_1",
2017-04-07T05:28:28.794505994Z      "establishment",
2017-04-07T05:28:28.79450804Z      "point_of_interest",
2017-04-07T05:28:28.794510959Z      "political"
2017-04-07T05:28:28.794514866Z    ]
2017-04-07T05:28:28.794517015Z  },
2017-04-07T05:28:28.794519126Z  ],
2017-04-07T05:28:28.79452157Z  "status": "OK"
2017-04-07T05:28:30.006223543Z }
2017-04-07T05:28:30.006223543Z 172.17.0.1 - - [07/Apr/2017 05:28:30] "GET /track?name=ashok HTTP/1.1"
200 -
```

Create a Service to expose the service outside:

Note we must use the `type=NodePort` because *minikube* doesn't support the *LoadBalancer* service. We can check if the service was exposed by listing services:

```
m-C02S23PLG8WM:docker-python-app aramar1$ kubectl expose deployment docker-python-app --type=NodePort
```

```
service "docker-python-app" exposed
```

```
m-C02S23PLG8WM:docker-python-app aramar1$
```

Get the services:

```
kubectl get svc
```

Services Dash board:

The screenshot shows the 'Services' page in the Kubernetes dashboard. The left sidebar is the same as the previous screenshot. The main panel displays a table of services. There are two services listed: 'docker-python-app' and 'kubernetes'. The 'docker-python-app' service is of type 'NodePort' and has an external endpoint. The 'kubernetes' service is of type 'ClusterIP' and has an external endpoint.

Name	Labels	Cluster IP	Internal endpoints	External endpoints
docker-python-app	run: docker-pytho...	10.0.0.44	docker-python-app:8... docker-python-app:3...	-
kubernetes	component: apise... provider: kubernet...	10.0.0.1	kubernetes:443 TCP kubernetes:0 TCP	-

```
m-C02S23PLG8WM:kubernetes aramar1$ kubectl get svc
```

NAME	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	10.0.0.1	<none>	443/TCP	8h
web	10.0.0.34	<nodes>	80:30940/TCP	3m

```
minikube stop
```