

## DEPLOYMENT OF APPLICATION IN IBM CLOUD

### ➤ UPLOAD THE IMAGE TO CONTAINER REGISTRY

```
kunals-mbp:web kunalmalhotra$ docker push registry.ng.bluemix.net/flask-node/app:latest
The push refers to repository [registry.ng.bluemix.net/flask-node/app]
d905410b27c1: Pushed
b96ded950728: Pushed
437e8db4a234: Pushed
ba9884d50644: Pushed
1989aa0f3739: Layer already exists
7bec9e49c283: Layer already exists
1172bcd1177f: Layer already exists
8eb4c3a69e64: Layer already exists
1fa8778eb779: Layer already exists
fa0c3f992cbd: Layer already exists
ce6466f43b11: Layer already exists
719d45669b35: Layer already exists
3b10514a95be: Layer already exists
latest: digest: sha256:5015254c21592b5ab08168707b74ddd763e97e80b59d9187afa2a80433b9d2ab size: 3061
kunals-mbp:web kunalmalhotra$
```

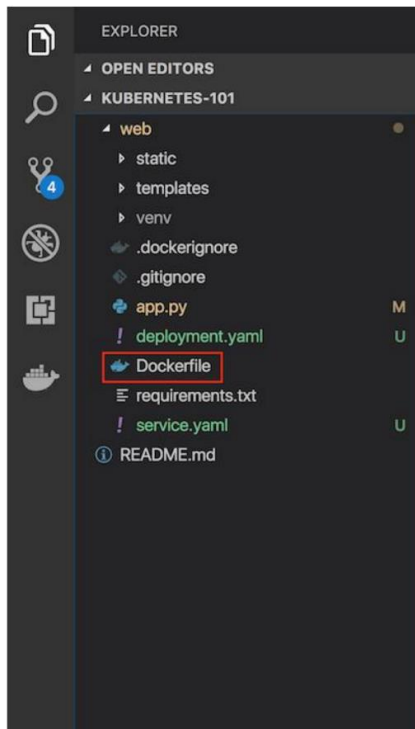
15. Verify that your image is in your private registry.

16. `ibmcloud cr image-list`

```
kunals-mbp:web kunalmalhotra$ ibmcloud cr image-list
Listing images...

REPOSITORY          TAG      DIGEST      NAMESPACE  CREATED    SIZE    SECURITY STATUS
registry.ng.bluemix.net/flask-node/app  latest  b721ad768feb flask-node  1 day ago  366 MB  3 Issues
OK
kunals-mbp:web kunalmalhotra$
```

## ➤ CONTAINERIZE THE APPLICATION



```

root@kali:~/kubernetes# docker build -t app:v0.1
Sending build context to Docker daemon 384.7kB
Step 1/6 : FROM python:2.7
--> 6c7e9397f7e
Step 2/6 : LABEL maintainer="Kunal Mishra, kunal.mishra@ibm.com"
--> 4d8633967fe
--> Using cache
--> ddbcf4359c
Step 3/6 : RUN apt-get update
--> Using cache
--> C9D5C34ade
Step 4/6 : CMD /app
--> 69f372799
Step 5/6 : WORKDIR /app
Running intermediate container f0c18099d2fe
--> ff0ecf73dc3
Step 6/6 : RUN pip install --requirements.txt
--> Running in b353ab0ab97
Collecting click==6.7 (from ~/.requirements.txt (line 2))
  Downloaded from https://files.pythonhosted.org/packages/7/c/d3/7087f273ad693c33662cc979ff1d3036af407f22ae479707f1c1c6-6-py2.py-none-any.whl (27kB)
    Collecting Flask==0.12 (from ~/.requirements.txt (line 2))
      Downloading https://files.pythonhosted.org/packages/76/82/76837f7ead53d242d34ac698334687df2ae69728cd0eb4b7/flask-1.0.2-py2.py-none-any.whl (91kB)
    Collecting Werkzeug==0.14 (from ~/.requirements.txt (line 3))
      Downloading https://files.pythonhosted.org/packages/d6/2b/6bbfd85ce566866806763272bc728c12188573265a4/Werkzeug-0.14.1.tar.gz (484kB)
    Collecting flask-wtf==0.14 (from ~/.requirements.txt (line 4))
      Downloading https://files.pythonhosted.org/packages/7e/f7/6e6bfcd05f72f66806763272bc728c12188573265a4/Werkzeug-0.14.1.tar.gz (484kB)
    Collecting flask-classful==0.1.2 (from ~/.requirements.txt (line 5))
      Downloading https://files.pythonhosted.org/packages/4a/76/32041816b16bf4b0882651766744225a90b9bfcb173/kwerkflask-1.0.0.tar.gz
    Collecting MarkupSafe==1.1 (from ~/.requirements.txt (line 6))
      Downloading https://files.pythonhosted.org/packages/5b/94/9375131de1214c8a7255250ca3676767683f697683f680464433563/MarkupSafe-1.1.1-py2.py-none-any.whl (352kB)
Building wheels for collected packages: Werkzeug, MarkupSafe
Running setup.py bdist_wheel for Werkzeug: finished with status 'done'
Running setup.py bdist_wheel for MarkupSafe: started
Running setup.py bdist_wheel for Werkzeug: finished with status 'done'
Running setup.py bdist_wheel for MarkupSafe: finished with status 'done'
Storing in directory /root/.cache/pip/wheels/23/96/26/5250ca36767683f697683f680464433563/Werkzeug-0.14.1-py2.py-none-any.whl
Successfully built Werkzeug MarkupSafe
Installing collected packages: Werkzeug, MarkupSafe, flask-wtf, Werkzeug, flask-classful
Successfully installed Flask-1.0.2 Jinja2-2.10 MarkupSafe-1.1 Werkzeug-0.14.1 click-6.7 Werkzeug-0.14
Cleaning intermediate container b2530c30db07
--> 66d63030db7
Step 7/6 : ENTRYPOINT ["python"]
--> Running in bc12518154
--> 79aefc8fc6
Running intermediate container hck18138161
Step 8/8 : CMD ["app.py"]
--> Running in a734ba3a6f8
Running intermediate container a78a03a0a6f
--> ddb6b33763
Successfully built appv0.1:latest
Successfully tagged app/v0.1

```

## Run your container locally and test

After you build your image succesfully, type: `docker run -d -p 5000:5000 app`

This command will create a container that contains all the application code and dependencies from the image and runs it locally.

```
kunelis-mtp-ssh:kunelis@jotrns:~$ docker run -d --p 5000-5000 app
1c2b8f86758e9d6000095262af339ed400eb82631c35d54160c616247
```

	COMMAND	CREATED	STATUS	PORTS	NAMES
kunelis-mtp-ssh:kunelis@jotrns:~\$ docker ps	nginx	ago	Up	0.0.0.0:5000->5000/tcp	compassionate_jalajay
1c2b8f86758e	"python app.py"	Less than a second ago	Up 5 seconds		

## ➤ DEPLOY IN KUBERNETES CLUSTER

The screenshot shows the IBM Cloud Clusters console for a cluster named 'mycluster-free'. The cluster is in the 'Preparing master, workers...' state and will expire in 30 days. The overview section displays the following status:

- Node status:** 1 of 1 Pending
- Add-on status:** 0 of 0 Normal
- Master status:** Unknown
- Ingress status:** Pending

The details section provides the following information:

- Cluster ID:** cdriief0uc6vs5gl4q0
- Version:** 1.24.8\_1544
- Infrastructure:** Classic
- Zones:** Milan 01
- Created:** 11/18/2022, 12:12 PM
- Resource group:** (empty)
- Image pull secrets:** Enable
- Image security enforcement:** Enable

The left sidebar shows the navigation menu with 'Overview' selected. The right sidebar contains a 'Help' dropdown menu with options: Log in to your cluster, Deploy your app, Expose your app, Add storage to your app, Connect integrations, Install add-ons, and Troubleshoot.

The screenshot shows the 'Worker nodes' tab for the 'mycluster-free' cluster. The table below lists the worker nodes:

Pool	Filter...	Search	Add				
	Name	Status	Worker pool	Zone	Private IP	Public IP	Version
	0000003d	Provision pending	default	Milan 01			--> 1.24.7_1543

The details section for the selected worker node provides the following information:

- ID:** kube-cdriief0uc6vs5gl4q0-myclusterfr-default-0000003d
- Status:** --
- Flavor:** Free - 2 vCPUs 4GB RAM
- Private VLAN:** 2218181
- Public VLAN:** 2218179

The bottom of the page shows the pagination: Items per page: 25, 1-1 of 1 item, 1 of 1 page.



