

🚦 K8S 14 SOLUTION

=====

Que 1 →

- Create 2 Public Docker Hub registries named `cloudethix_master_nginx_yourname` & `cloudethix_release_nginx_yourname`.
- Clone below repository on your system.

`https://github.com/zembutsu/docker-sample-nginx.git`

- Initialize a local repository & copy the code from above repo to your local

repository in master branch and then create below branches.

release

main

hotfix

- Once code is copied to local repository, from master branch update the

`index.html` and add word "Cloudethix Master Branch Nginx" and build the docker image & add meaningful tags and push to Docker Hub registry `cloudethix_master_nginx_yourname`.

- Also from release branch update the `index.html` and add word "Cloudethix

Release Branch Nginx" and build the docker image & add meaningful tags and push to Docker Hub registry `cloudethix_release_nginx_yourname`.

- Once Images are copied to Docker hub registries, switch to the main branch.

- In main branch create directory named `kube/clusterIP` & inside kube directory create file named `master_pod.yaml` with pod name `master_nginx` &

with label `master_nginx` & add image that you have pushed in Docker Hub registry `cloudethix_master_nginx_yourname`.

- Also create a file `release_pod.yaml` with pod name `release_nginx` & with

label `release_nginx` & add image that you have pushed in Docker Hub registry `cloudethix_release_nginx_yourname`.

- Create a file called `cluster_ip-service.yaml` with service name

cloudethix_clusterip and with Type clusterIP.

- Then, select the pod with label release_nginx in service.
- Create all these three resources in your k8s cluster.
- Now, access master_nginx pod shell & curl the master_nginx pod & check

the result.

- Also try to curl release_nginx pod with DNS name & check the result.
- Then curl the clusterip service with its name and check the result.
- Finally, create a GITHUB remote repository named cloudethix-k8s-yourname and push all the branches to the remote repository.
- Take all screenshots and create a well formatted document.

=====

➤ SOLUTION

```
root@DESKTOP-800G2HF:KUBERNETES-ASSIGNMENTS# mkdir K8S-Questions-Assignment
```

```
root@DESKTOP-800G2HF:KUBERNETES-ASSIGNMENTS# cd K8S-Questions-Assignment/
```

```
root@DESKTOP-800G2HF:K8S-Questions-Assignment# ll
```

```
total 0
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:00 ./
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:00 ../
```

```
root@DESKTOP-800G2HF:K8S-Questions-Assignment# ll
```

```
total 0
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:00 ./
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:00 ../
```

```
root@DESKTOP-800G2HF:K8S-Questions-Assignment# mkdir Question_01
```

```
root@DESKTOP-800G2HF:K8S-Questions-Assignment# cd Question_01/
```

```
root@DESKTOP-800G2HF:Question_01# git clone  
git@github.com:zembutsu/docker-sample-nginx.git
```

```
Cloning into 'docker-sample-nginx'...
```

```
remote: Enumerating objects: 22, done.
```

```
remote: Counting objects: 100% (12/12), done.
```

```
remote: Compressing objects: 100% (6/6), done.
```

```
remote: Total 22 (delta 7), reused 6 (delta 6), pack-reused 10
```

```
Receiving objects: 100% (22/22), done.
```

Resolving deltas: 100% (7/7), done.

```
root@DESKTOP-800G2HF:Question_01# ll
```

```
total 0
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:01 ./
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:01 ../
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:01 docker-sample-nginx/
```

```
root@DESKTOP-800G2HF:Question_01# cd docker-sample-nginx/
```

```
root@DESKTOP-800G2HF:docker-sample-nginx# ll
```

```
total 4
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:01 ./
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:01 ../
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:01 .git/
```

```
-rwxrwxrwx 1 asfiya asfiya 95 Feb 21 12:01 Dockerfile*
```

```
-rwxrwxrwx 1 asfiya asfiya 1084 Feb 21 12:01 LICENSE*
```

```
-rwxrwxrwx 1 asfiya asfiya 73 Feb 21 12:01 README.md*
```

```
-rwxrwxrwx 1 asfiya asfiya 286 Feb 21 12:01 default.conf*
```

```
-rwxrwxrwx 1 asfiya asfiya 103 Feb 21 12:01 index.html*
```

```
root@DESKTOP-800G2HF:docker-sample-nginx# mkdir local-repository
```

```
root@DESKTOP-800G2HF:docker-sample-nginx# cd local-repository/
```

```
root@DESKTOP-800G2HF:local-repository# cp -pr ../Dockerfile .
```

```
root@DESKTOP-800G2HF:local-repository# cp -pr ../default.conf .
```

```
root@DESKTOP-800G2HF:local-repository# cp -pr ../index.html .
```

```
root@DESKTOP-800G2HF:local-repository# ll
```

```
total 0
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:10 ./
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:08 ../
```

```
-rwxrwxrwx 1 asfiya asfiya 95 Feb 21 12:01 Dockerfile*
```

```
-rwxrwxrwx 1 asfiya asfiya 286 Feb 21 12:01 default.conf*
```

```
-rwxrwxrwx 1 asfiya asfiya 103 Feb 21 12:01 index.html*
```

```
root@DESKTOP-800G2HF:local-repository# git branch release
```

```
root@DESKTOP-800G2HF:local-repository# git branch main
```

```
root@DESKTOP-800G2HF:local-repository# git branch hotfix
```

```
root@DESKTOP-800G2HF:local-repository# ll
```

```
total 0
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:10 ./
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:08 ../
```

```
-rwxrwxrwx 1 asfiya asfiya 95 Feb 21 12:01 Dockerfile*
```

```
-rwxrwxrwx 1 asfiya asfiya 286 Feb 21 12:01 default.conf*
```

```
-rwxrwxrwx 1 asfiya asfiya 103 Feb 21 12:01 index.html*
```

```
root@DESKTOP-800G2HF:local-repository# vim Dockerfile
```

```
root@DESKTOP-800G2HF:local-repository# vim index.html
```

```
root@DESKTOP-800G2HF:local-repository# cat index.html
```

```
<html>
```

```
<body>
```

```
    <h1>Host: <!--#echo var="HOSTNAME" --></h1>
```

```
    Version: 1.1
```

```
    Cloudethix Master Branch Nginx
```

```
</body>
```

```
</html>
```

```
root@DESKTOP-800G2HF:local-repository# docker image build -t  
asfiyask/cloudethix_master_nginx_asfiya:v1.0 .
```

```
[+] Building 18.3s (9/9) FINISHED
```

```
docker:default
```

```
=> [internal] load build definition from Dockerfile
```

```
0.1s
```

```
=> => transferring dockerfile: 132B
```

```
0.0s
```

```
=> [internal] load metadata for docker.io/library/nginx:alpine
```

```
13.5s
```

```
=> [auth] library/nginx:pull token for registry-1.docker.io
```

```
0.0s
```

```
=> [internal] load .dockerignore
```

```
0.0s
```

```
=> => transferring context: 2B
```

```
0.0s
```

```
=> [1/3] FROM
```

```
docker.io/library/nginx:alpine@sha256:6a2f8b28e45c4adea04ec207a251fd4a2df03ddc930f782  
af51e315ebc76e9a9 4.1s
```

```
=> => resolve
```

```
docker.io/library/nginx:alpine@sha256:6a2f8b28e45c4adea04ec207a251fd4a2df03ddc930f782  
af51e315ebc76e9a9 0.0s
```

```
=> => sha256:6a2f8b28e45c4adea04ec207a251fd4a2df03ddc930f782af51e315ebc76e9a9 8.71kB  
/ 8.71kB 0.0s
```

```
=> => sha256:cb0953165f59b5cf2227ae979a49a2284956d997fad4ed7a338eebc6aef3e70b 2.50kB  
/ 2.50kB 0.0s
```

```
=> => sha256:6913ed9ec8d009744018c1740879327fe2e085935b2cce7a234bf05347b670d7  
11.74kB / 11.74kB 0.0s
```

```
=> => sha256:619be1103602d98e1963557998c954c892b3872986c27365e9f651f5bc27cab8 3.40MB  
/ 3.40MB 0.6s
```

```
=> => sha256:018b9065ed0dfedff48bbd11f6014960bb496e71c395f772bfad123ab33a1800 1.90MB  
/ 1.90MB 0.9s
```

```
=> => sha256:c3ea3344e711fd7111dee02f17deebceb725ed1d0ee998f7fb472114dc1399ce 629B /  
629B 0.6s
```

```
=> => extracting
```

```
sha256:619be1103602d98e1963557998c954c892b3872986c27365e9f651f5bc27cab8
```

```
0.2s
```

```
=> => sha256:a101c9a82b88a3fa561030af162d98a130ca3bc0501b2e70594410dd426f2c9b 393B /  
393B 1.2s
```

```
=> => sha256:c7059f3102784cd05dc96fff74a52bce9fa50fea724ece08748507fa3455999b 956B /  
956B 1.0s
```

```

=> => extracting
sha256:018b9065ed0dfedff48bbd11f6014960bb496e71c395f772bfad123ab33a1800
0.4s
=> => sha256:d6a456492aaa4c003389fec3da0939f31c505232fcf1925db314815a196c444f 1.21kB
/ 1.21kB 1.4s
=> => sha256:e1c681003a03fff277ecf90fccf526881bcc2e006c9e371b58f45680d54c1954 1.40kB
/ 1.40kB 1.4s
=> => sha256:a85ccd8c07bd7090e8a37ab878413b035a370e872367b145a0c0aaaaf60ccbdf
12.65MB / 12.65MB 2.9s
=> => extracting
sha256:c3ea3344e711fd7111dee02f17deebceb725ed1d0ee998f7fb472114dc1399ce
0.0s
=> => extracting
sha256:c7059f3102784cd05dc96fff74a52bce9fa50fea724ece08748507fa3455999b
0.0s
=> => extracting
sha256:a101c9a82b88a3fa561030af162d98a130ca3bc0501b2e70594410dd426f2c9b
0.0s
=> => extracting
sha256:d6a456492aaa4c003389fec3da0939f31c505232fcf1925db314815a196c444f
0.0s
=> => extracting
sha256:e1c681003a03fff277ecf90fccf526881bcc2e006c9e371b58f45680d54c1954
0.0s
=> => extracting
sha256:a85ccd8c07bd7090e8a37ab878413b035a370e872367b145a0c0aaaaf60ccbdf
0.7s
=> [internal] load build context
0.1s
=> => transferring context: 491B
0.0s
=> [2/3] COPY default.conf /etc/nginx/conf.d/
0.3s
=> [3/3] COPY index.html /usr/share/nginx/html/
0.1s
=> exporting to image
0.1s
=> => exporting layers
0.1s
=> => writing image
sha256:4ad9a9dadf9cbb6c37dd7a51b60d58c813f95655023992424d24767dea45ae30
0.0s
=> => naming to docker.io/asfiyask/cloudethix_master_nginx_asfiya:v1.0
0.0s

```

```

root@DESKTOP-800G2HF:local-repository# docker image push
asfiyask/cloudethix_master_nginx_asfiya:v1.0

```

The push refers to repository
[docker.io/asfiyask/cloudethix_master_nginx_asfiya]

```

b236441bb395: Pushed
b397c7fc2d49: Pushed
667a247707f0: Mounted from library/nginx
d8527026595f: Mounted from library/nginx
2593b08e5428: Mounted from library/nginx
9909978d630d: Mounted from library/nginx
c5140fc719dd: Mounted from library/nginx
3137f8f0c641: Mounted from asfiyask/nginx
718db50a47c0: Mounted from library/nginx
aedc3bda2944: Mounted from library/nginx

```

```
v1.0: digest: sha256:08804e31ee61176238fc2c94690c2d1dedad7d389a0d8d6ecc1f29afcb3021a4
size: 2403
```

```
root@DESKTOP-800G2HF:local-repository# git switch release
```

```
Switched to branch 'release'
```

```
root@DESKTOP-800G2HF:local-repository# git branch
```

```
hotfix
```

```
main
```

```
master
```

```
* release
```

```
root@DESKTOP-800G2HF:local-repository# ll
```

```
total 0
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:18 ./
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:08 ../
-rwxrwxrwx 1 asfiya asfiya  95 Feb 21 12:17 Dockerfile*
-rwxrwxrwx 1 asfiya asfiya 286 Feb 21 12:01 default.conf*
-rwxrwxrwx 1 asfiya asfiya 125 Feb 21 12:18 index.html*
```

```
root@DESKTOP-800G2HF:local-repository# cat index.html
```

```
<html>
<body>
    <h1>Host: <!--#echo var="Cloudethix Master Branch Nginx" --></h1>
    Version: 1.1
</body>
</html>
```

```
root@DESKTOP-800G2HF:local-repository# vim index.html
```

```
root@DESKTOP-800G2HF:local-repository# cat index.html
```

```
<html>
<body>
    <h1>Host: <!--#echo var="Cloudethix Release Branch Nginx" --></h1>
    Version: 1.1
</body>
</html>
```

```
root@DESKTOP-800G2HF:local-repository# docker image build -t
asfiyask/cloudethix_release_nginx_asfiya:v1.1 .
```

```
[+] Building 1.8s (9/9) FINISHED
docker:default
=> [internal] load build definition from Dockerfile
0.0s
=> => transferring dockerfile: 132B
0.0s
=> [internal] load metadata for docker.io/library/nginx:alpine
1.5s
=> [auth] library/nginx:pull token for registry-1.docker.io
0.0s
=> [internal] load .dockerignore
0.0s
```

```

=> => transferring context: 2B
0.0s
=> [1/3] FROM
docker.io/library/nginx:alpine@sha256:6a2f8b28e45c4adea04ec207a251fd4a2df03ddc930f782
af51e315ebc76e9a9 0.0s
=> [internal] load build context
0.0s
=> => transferring context: 199B
0.0s
=> CACHED [2/3] COPY default.conf /etc/nginx/conf.d/
0.0s
=> [3/3] COPY index.html /usr/share/nginx/html/
0.1s
=> exporting to image
0.1s
=> => exporting layers
0.0s
=> => writing image
sha256:5097c6e1faa1a75ae34e1b6f6869d3ca8c8b70a26067b7bc15c72e6b4f9e4a97
0.0s
=> => naming to docker.io/asfiyask/cloudethix_release/nginx_asfiya:v1.1
0.0s
root@DESKTOP-800G2HF:local-repository# docker image push
asfiyask/cloudethix_release/nginx_asfiya:v1.1

```

The push refers to repository
[docker.io/asfiyask/cloudethix_release/nginx_asfiya]

```

64a0eb9b9a77: Pushed
b397c7fc2d49: Mounted from asfiyask/cloudethix_master/nginx_asfiya
667a247707f0: Mounted from asfiyask/cloudethix_master/nginx_asfiya
d8527026595f: Mounted from asfiyask/cloudethix_master/nginx_asfiya
2593b08e5428: Mounted from asfiyask/cloudethix_master/nginx_asfiya
9909978d630d: Mounted from asfiyask/cloudethix_master/nginx_asfiya
c5140fc719dd: Mounted from asfiyask/cloudethix_master/nginx_asfiya
3137f8f0c641: Mounted from asfiyask/cloudethix_master/nginx_asfiya
718db50a47c0: Mounted from asfiyask/cloudethix_master/nginx_asfiya
aedc3bda2944: Mounted from asfiyask/cloudethix_master/nginx_asfiya
v1.1: digest: sha256:d3e38768d592b6f995e275a1592a9edc99cab6f8e828e9bd481506af9382ac43
size: 2403

```

```

root@DESKTOP-800G2HF:local-repository# git switch main

```

Switched to branch 'main'

```

root@DESKTOP-800G2HF:local-repository#

```

```

root@DESKTOP-800G2HF:local-repository# git branch

```

```

    hotfix

```

```

* main

```

```

    master

```

```

    release

```

```

root@DESKTOP-800G2HF:local-repository# mkdir kube

```

```
root@DESKTOP-800G2HF:local-repository# cd kube/
root@DESKTOP-800G2HF:kube# touch master_pod.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: master-nginx
  labels:
    app: master-nginx
spec:
  containers:
  - name: master-nginx-container
    image: asfiyask/cloudethix_master_nginx_asfiya:v2
    ports:
    - containerPort: 80
```

```
root@DESKTOP-800G2HF:kube# touch release_pod.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: release-nginx
  labels:
    app: release-nginx
spec:
  containers:
  - name: release-nginx-container
    image: asfiyask/cloudethix_release_nginx_asfiya:v2
    ports:
    - containerPort: 80
```

```
root@DESKTOP-800G2HF:local-repository# mkdir clusterIP
```

```
root@DESKTOP-800G2HF:local-repository# cd clusterIP/
```

```
root@DESKTOP-800G2HF:clusterIP# ll
```

```
total 0
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:46 ./
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:45 ../
```

```
-rwxrwxrwx 1 asfiya asfiya 225 Feb 21 12:43 cluster_ip-service.yaml*
```

```
# NODE2 SERVICE.YAML
apiVersion: v1
kind: Service
metadata:
  name: cloudethix-clusterip
spec:
```



```

ports:
  - port: 80
    protocol: TCP
    targetPort: 80
selector:
  app: release-nginx
type: ClusterIP

```

Apply complete! Resources: 15 added, 0 changed, 0 destroyed.

Outputs:

```

cluster_nodes = [
  {
    "name" = "master"
    "private_ip" = "172.31.33.64"
    "public_ip" = "65.0.165.194"
    "subnet_id" = "subnet-07fa90d35a2e2f0d5"
  },
  {
    "name" = "worker-0"
    "private_ip" = "172.31.46.92"
    "public_ip" = "65.0.168.192"
    "subnet_id" = "subnet-07fa90d35a2e2f0d5"
  },
  {
    "name" = "worker-1"
    "private_ip" = "172.31.40.201"
    "public_ip" = "13.234.77.46"
    "subnet_id" = "subnet-07fa90d35a2e2f0d5"
  },
]
vpc_id = "vpc-08cc066fb89d9b742"

```

```
root@DESKTOP-800G2HF:k8sCluster_kubeadm_terraform# realpath assured-bull.conf
```

```
/mnt/c/Users/DELL/OneDrive/Desktop/CLOUDETHIX/KUBERNETES/k8sCluster_kubeadm_terraform/assured-bull.conf
```

```
root@DESKTOP-800G2HF:k8sCluster_kubeadm_terraform# export
KUBECONFIG=/mnt/c/Users/DELL/OneDrive/Desktop/CLOUDETHIX/KUBERNETES/k8sCluster_kubeadm_terraform/assured-bull.conf
```

```
root@DESKTOP-800G2HF:clusterIP# kgn
```

NAME	STATUS	ROLES	AGE	VERSION
master	Ready	control-plane	110s	v1.28.2
worker-0	Ready	<none>	97s	v1.28.2
worker-1	Ready	<none>	97s	v1.28.2

```
root@DESKTOP-800G2HF:k8sCluster_kubeadm_terraform# kubectl apply -f calico.yaml
```

```
root@DESKTOP-800G2HF:kube# k apply -f .
```

```
pod/master-nginx created
```

pod/release-nginx created

root@DESKTOP-800G2HF:kube# cd ../clusterIP/

root@DESKTOP-800G2HF:clusterIP# k apply -f .

service/cloudethix-clusterip created

root@DESKTOP-800G2HF:clusterIP# cd -

/mnt/c/Users/DELL/OneDrive/Desktop/CLOUDETHIX/KUBERNETES/KUBERNETES-
ASSIGNMENTS/K8S-Questions-Assignment/Question_01/docker-sample-
nginx/local-repository/kube

root@DESKTOP-800G2HF:kube# ll

```
total 0
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 12:46 ./
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 13:47 ../
-rwxrwxrwx 1 asfiya asfiya 241 Feb 21 13:52 master_pod.yaml*
-rwxrwxrwx 1 asfiya asfiya 244 Feb 21 13:49 release_pod.yaml*
```

root@DESKTOP-800G2HF:kube# kgp

NAME	READY	STATUS	RESTARTS	AGE
master-nginx	1/1	Running	0	19s
release-nginx	1/1	Running	0	19s

root@DESKTOP-800G2HF:kube# k exec -it master-nginx -- /bin/sh

```
/ # curl localhost
<html>
<body>
    <h1>Host: master-nginx</h1>
    Version: 1.1
    Cloudethix Master Branch Nginx
</body>
</html>
/ #
```

root@DESKTOP-800G2HF:kube# kgs

NAME	AGE	TYPE	CLUSTER-IP	EXTERNAL-IP
cloudethix-clusterip	2m	ClusterIP	10.101.120.181	<none>

root@DESKTOP-800G2HF:kube# k exec -it release-nginx -- /bin/sh

```
/ # curl localhost
<html>
<body>
    <h1>Host: release-nginx</h1>
    Version: 1.1
```

Cloudethix release Branch Nginx

</body>

</html>

```
root@DESKTOP-800G2HF:kube# k exec -it master-nginx -- /bin/sh
```

```
/ # curl cloudethix-clusterip
```

<html>

<body>

<h1>Host: release-nginx</h1>

Version: 1.1

Cloudethix release Branch Nginx

</body>

</html>

```
root@DESKTOP-800G2HF:docker-sample-nginx# git remote -v
```

```
root@DESKTOP-800G2HF:docker-sample-nginx# git remote add origin  
git@github.com:ASFIASHAIKH/cloudethix-k8s-asfiya.git
```

```
root@DESKTOP-800G2HF:docker-sample-nginx# git remote -v
```

```
origin git@github.com:ASFIASHAIKH/cloudethix-k8s-asfiya.git (fetch)
```

```
origin git@github.com:ASFIASHAIKH/cloudethix-k8s-asfiya.git (push)
```

```
root@DESKTOP-800G2HF:docker-sample-nginx# git add local-repository
```

```
root@DESKTOP-800G2HF:docker-sample-nginx# git commit -m "Add  
Assignment_05 Question-1 File"
```

```
[master 5513230] Add Assignment_05 Question-1 File
```

```
6 files changed, 67 insertions(+)
```

```
create mode 100644 local-repository/Dockerfile
```

```
create mode 100644 local-repository/clusterIP/cluster_ip-service.yaml
```

```
create mode 100644 local-repository/default.conf
```

```
create mode 100644 local-repository/index.html
```

```
create mode 100644 local-repository/kube/master_pod.yaml
```

```
create mode 100644 local-repository/kube/release_pod.yaml
```

```
root@DESKTOP-800G2HF:docker-sample-nginx# git branch
```

```
hotfix
```

```
main
```

```
* master
```

Release

```
root@DESKTOP-800G2HF:docker-sample-nginx# git push --all
```

```
Enumerating objects: 31, done.
Counting objects: 100% (31/31), done.
Delta compression using up to 4 threads
Compressing objects: 100% (24/24), done.
Writing objects: 100% (31/31), 4.07 KiB | 47.00 KiB/s, done.
Total 31 (delta 8), reused 21 (delta 7), pack-reused 0
remote: Resolving deltas: 100% (8/8), done.
remote:
remote: Heads up! The branch 'main' that you pushed to was renamed to 'Remote-Main-Branch'.
remote:
To github.com:ASFIASHAIKH/cloudethix-k8s-asfiya.git
 * [new branch]      hotfix -> hotfix
 * [new branch]      main -> main
 * [new branch]      master -> master
 * [new branch]      release -> release
=====
```

Que 2 →

- In the main branch of your local repository create a directory kube/NodePort.

- Create below files from below url. Please make sure you will create

NodePort service with port 30008 instead of loadbalancer.

<https://kubernetes.io/docs/tasks/access-application-cluster/connec>

ting-frontend-backend/.

backend-deployment.yaml

backend-service.yaml

frontend-deployment.yaml

frontend-NodePort-service.yaml

- Once files are created , create all the resources in your k8s cluster.

- Access all public ips with port 30008 in the browser and then check the

result.

- Finally, push all the latest code to the remote repository.

=====

➤ SOLUTION

```
root@DESKTOP-800G2HF:docker-sample-nginx# git branch
```

```
hotfix
```

```
main
```

```
* master
```

```
release
```

```
root@DESKTOP-800G2HF:docker-sample-nginx# git switch main
```

```
Switched to branch 'main'
```

```
root@DESKTOP-800G2HF:docker-sample-nginx# git branch
```

```
hotfix
```

```
* main
```

```
master
```

```
release
```

```
root@DESKTOP-800G2HF:kube-NodePort# touch backend-service.yaml
```

```
root@DESKTOP-800G2HF:kube-NodePort# touch frontend-deployment.yaml
```

```
root@DESKTOP-800G2HF:kube-NodePort# touch frontend-NodePort-  
service.yaml
```

```
root@DESKTOP-800G2HF:kube-NodePort# ll
```

```
total 4
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 14:56 ./
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 14:43 ../
```

```
-rwxrwxrwx 1 asfiya asfiya 488 Feb 21 15:02 backend-deployment.yaml*
```

```
-rwxrwxrwx 1 asfiya asfiya 187 Feb 21 14:58 backend-service.yaml*
```

```
-rwxrwxrwx 1 asfiya asfiya 228 Feb 21 15:01 frontend-NodePort-service.yaml*
```

```
-rwxrwxrwx 1 asfiya asfiya 538 Feb 21 15:02 frontend-deployment.yaml*
```

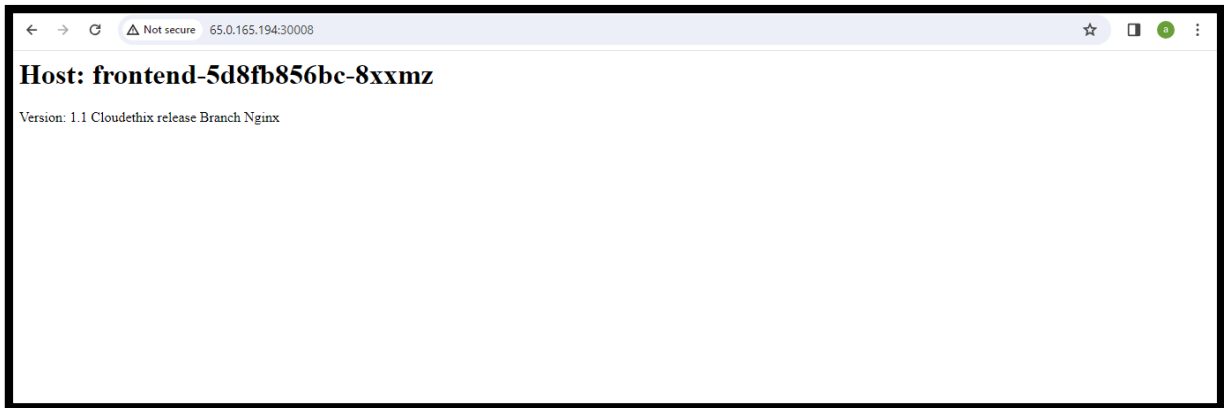
```
root@DESKTOP-800G2HF:kube-NodePort# kgp
```

NAME	READY	STATUS	RESTARTS	AGE
backend-68bdd7b78f-ch954	1/1	Running	0	12s
backend-68bdd7b78f-kd67p	1/1	Running	0	12s
backend-68bdd7b78f-thkfw	1/1	Running	0	12s
frontend-5d8fb856bc-8xxmz	1/1	Running	0	12s

```
root@DESKTOP-800G2HF:kube-NodePort# kgs
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP
PORT(S)	AGE		

frontend		NodePort	10.108.168.23	<none>
80:30008/TCP	5m17s			
hello		ClusterIP	10.108.182.109	<none>
80/TCP	5m17s			
kubernetes		ClusterIP	10.96.0.1	<none>
443/TCP	129m			



=====

Que 3 →

- Create any 2 pods and assign them to different worker nodes with nodeName property.

=====

➤ SOLUTION

```
# pod-01
apiVersion: v1
kind: Pod
metadata:
  name: pod1
spec:
  nodeName: worker-0
  containers:
  - name: pod1-container
    image: asfiyask/nginx:v4

---
# pod-02
apiVersion: v1
kind: Pod
metadata:
  name: pod2
```

```
spec:
  nodeName: worker-1
  containers:
  - name: pod2-container
    image: asfiyask/nginx:v5
```

```
root@DESKTOP-800G2HF:pod-configs# kgp
```

NAME	READY	STATUS	RESTARTS	AGE
pod1	1/1	Running	0	10s
pod2	1/1	Running	0	10s

```
root@DESKTOP-800G2HF:pod-configs# kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED	NODE
pod1	1/1	Running	0	2m22s	10.108.43.18	worker-0	<none>	<none>
pod2	1/1	Running	0	2m22s	10.111.158.76	worker-1	<none>	<none>

Que 4 → ● Label both worker nodes such as worker-0 node as cloudethix-k8s-00 & worker-1 node as cloudethix-k8s-01

● Once nodes are labeled, create pod00.yaml file and schedule the pod on worker-0 node with nodeSelector property. Also create one more file named pod01.yaml & schedule the pod on worker-1 node.

➤ SOLUTION

```
root@DESKTOP-800G2HF:local-repository# cd Label-Nodes/
```

```
root@DESKTOP-800G2HF:Label-Nodes# ll
```

```
total 0
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 16:29 ./
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 16:50 ../
```

```
-rwxrwxrwx 1 asfiya asfiya 205 Feb 21 16:48 label-nodes.yaml*
```

```
# Label pod worker-0
apiVersion: v1
kind: Node
metadata:
  name: worker-0
  labels:
    cloudethix-k8s-00: "true"

---
# Label pod worker-1
apiVersion: v1
kind: Node
metadata:
  name: worker-1
  labels:
    cloudethix-k8s-01: "true"
```

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-00
spec:
  containers:
    - name: container-00
      image: asfiyask/nginx:v1
  nodeSelector:
    cloudethix-k8s-00: "true"
```

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-01
spec:
  containers:
    - name: container-01
      image: asfiyask/nginx:v2
  nodeSelector:
    cloudethix-k8s-01: "true"
```

```
root@DESKTOP-800G2HF:Label-Nodes# k apply -f .
```

```
node/worker-0 unchanged
```

```
node/worker-1 unchanged
```



```
pod/pod-00 created
```

```
pod/pod-01 created
```

```
root@DESKTOP-800G2HF:Label-Nodes# kgp
```

NAME	READY	STATUS	RESTARTS	AGE
pod-00	1/1	Running	0	14s
pod-01	1/1	Running	0	14s

```
root@DESKTOP-800G2HF:Label-Nodes# kubectl describe pod pod-01
```

```
Volumes:
  kube-api-access-r49q2:
    Type:              Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:      kube-root-ca.crt
    ConfigMapOptional:  <nil>
    DownwardAPI:        true
QoS Class:           BestEffort
Node-Selectors:      cloudethix-k8s-01=true
Tolerations:         node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                     node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason      Age   From          Message
  ----     -
  Normal   Scheduled   5m49s default-scheduler Successfully assigned default/pod-01 to worker-1
  Normal   Pulling     5m48s kubelet        Pulling image "asfiyask/nginx:v2"
  Normal   Pulled      5m43s kubelet        Successfully pulled image "asfiyask/nginx:v2" in 4.939s (4.939s including waiting)
  Normal   Created     5m43s kubelet        Created container container-01
  Normal   Started     5m43s kubelet        Started container container-01
```

```
root@DESKTOP-800G2HF:Label-Nodes# kubectl describe pod pod-00
```

```
Volumes:
  kube-api-access-mbnb5:
    Type:              Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:      kube-root-ca.crt
    ConfigMapOptional:  <nil>
    DownwardAPI:        true
QoS Class:           BestEffort
Node-Selectors:      cloudethix-k8s-00=true
Tolerations:         node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                     node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason      Age   From          Message
  ----     -
  Normal   Scheduled   9m33s default-scheduler Successfully assigned default/pod-00 to worker-0
  Normal   Pulling     9m32s kubelet        Pulling image "asfiyask/nginx:v1"
  Normal   Pulled      9m27s kubelet        Successfully pulled image "asfiyask/nginx:v1" in 4.962s (4.962s including waiting)
  Normal   Created     9m27s kubelet        Created container container-00
  Normal   Started     9m27s kubelet        Started container container-00
```

```
=====
Que 5 →
```

- Clone the below repo locally & create DaemonSet from directory DaemonSet101. <https://github.com/collabnix/kubelabs>

```
=====
SOLUTION
```

```
root@DESKTOP-800G2HF:DaemonSet# ll
```

```
total 0
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 17:22 ./
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 17:19 ../
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 17:26 kubelabs/
```

```
root@DESKTOP-800G2HF:DaemonSet101#
```

```
root@DESKTOP-800G2HF:DaemonSet101# ll
```

```
total 8
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 17:26 ./
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 17:26 ../
-rwxrwxrwx 1 asfiya asfiya 7040 Feb 21 17:26 README.md*
-rwxrwxrwx 1 asfiya asfiya 394 Feb 21 17:26 daemonset.yml*
```

```
root@DESKTOP-800G2HF:DaemonSet101# kubectl apply -f daemonset.yml
daemonset.apps/prometheus-daemonset created
```

```
root@DESKTOP-800G2HF:DaemonSet101# kubectl get daemonsets/prometheus-
daemonset
```

NAME	DESIRED	CURRENT	READY	UP-TO-DATE
AVAILABLE	NODE SELECTOR	AGE		
prometheus-daemonset	2	2	2	2
<none>	23s			

```
root@DESKTOP-800G2HF:DaemonSet101# kubectl describe
daemonset/prometheus-daemonset
```

```
Name:          prometheus-daemonset
Selector:      name=prometheus-exporter,tier=monitoring
Node-Selector: <none>
Labels:        <none>
Annotations:   deprecated.daemonset.template.generation: 1
Desired Number of Nodes Scheduled: 2
Current Number of Nodes Scheduled: 2
Number of Nodes Scheduled with Up-to-date Pods: 2
```

Number of Nodes Scheduled with Available Pods: 2

Number of Nodes Misscheduled: 0

Pods Status: 2 Running / 0 Waiting / 0 Succeeded / 0 Failed

Pod Template:

Labels: name=prometheus-exporter
tier=monitoring

Containers:

prometheus:

Image: prom/node-exporter

Port: 80/TCP

Host Port: 0/TCP

Environment: <none>

Mounts: <none>

Volumes: <none>

Events:

Type	Reason	Age	From	Message
----	-----	----	----	-----
Normal	SuccessfulCreate	38s	daemonset-controller	Created pod: prometheus-daemonset-2pn26
Normal	SuccessfulCreate	38s	daemonset-controller	Created pod: prometheus-daemonset-thskb

root@DESKTOP-800G2HF:DaemonSet101# kgn

NAME	STATUS	ROLES	AGE	VERSION
master	Ready	control-plane	5h25m	v1.28.2
worker-0	Ready	<none>	96m	v1.28.2
worker-1	Ready	<none>	96m	v1.28.2

=====

Que 6 → ● Create a static pod with name cloudethix-static in your k8s cluster. Refer below link.
<https://kubernetes.io/docs/tasks/configure-pod-container/static-pod/>

=====

SOLUTION

```
root@DESKTOP-800G2HF:local-repository# mkdir Static-Pod
```

```
root@DESKTOP-800G2HF:local-repository# cd Static-Pod/
```

```
root@DESKTOP-800G2HF:Static-Pod# ll
```

```
total 0
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 18:30 ./
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 18:30 ../
```

```
root@DESKTOP-800G2HF:manifests# ssh -i ~/.ssh/id_rsa  
ubuntu@65.0.168.192
```

```
The authenticity of host '65.0.168.192 (65.0.168.192)' can't be  
established.
```

```
ED25519 key fingerprint is  
SHA256:aOSwPVQgJAjxq7E5Ut/F3rQ803I/sTb+sqeQOPF61B4.
```

```
This key is not known by any other names
```

```
Are you sure you want to continue connecting  
(yes/no/[fingerprint])? yes
```

```
Warning: Permanently added '65.0.168.192' (ED25519) to the list  
of known hosts.
```

```
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1103-aws x86_64)
```

- * Documentation: <https://help.ubuntu.com>
- * Management: <https://landscape.canonical.com>
- * Support: <https://ubuntu.com/advantage>

```
System information as of Wed Feb 21 13:23:07 UTC 2024
```

```
System load: 0.0          Users logged in: 0  
Usage of /: 51.8% of 7.57GB IP address for eth0:  
172.31.46.92
```

Memory usage: 26%
172.17.0.1

IP address for docker0:

Swap usage: 0%
10.108.43.21

IP address for tunl0:

Processes: 146

```
ubuntu@ip-172-31-46-92:~$ ps -ef | grep kubel
```

```
root      5679      1  0 07:29 ?           00:03:20 /usr/bin/kubelet  
--bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --  
kubeconfig=/etc/kubernetes/kubelet.conf --  
config=/var/lib/kubelet/config.yaml --container-runtime-  
endpoint=unix:///var/run/containerd/containerd.sock --hostname-  
override=worker-0 --pod-infra-container-  
image=registry.k8s.io/pause:3.9
```

```
ubuntu@ip-172-31-46-92:/etc/kubernetes/manifests$ sudo su
```

```
root@ip-172-31-46-92:/etc/kubernetes/manifests# vim static-  
web.yaml
```

```
root@ip-172-31-46-92:/etc/kubernetes/manifests# cat static-  
web.yaml
```

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
  name: static-web
```

```
  labels:
```

```
    role: myrole
```

```
spec:
```

```
  containers:
```

```
    - name: web
```

```
      image: nginx
```

```
      ports:
```

```
        - name: web
```

containerPort: 80

protocol: TCP

root@ip-172-31-46-92:/etc/kubernetes/manifests# exit

ubuntu@ip-172-31-46-92:/etc/kubernetes/manifests\$ logout

Connection to 65.0.168.192 closed.

root@DESKTOP-800G2HF:manifests# kgp

NAME	READY	STATUS	RESTARTS	AGE
static-web-worker-0	1/1	Running	0	57s

=====

Que 7 →

- Install Kubectl & kubens in your k8s cluster

✓ INSTALL KUBECTX-KUBENS

sudo apt-get update

sudo snap install kubectl --classic

chmod +x kubectl kubens

sudo mv kubectl kubens /usr/local/bin

snap list

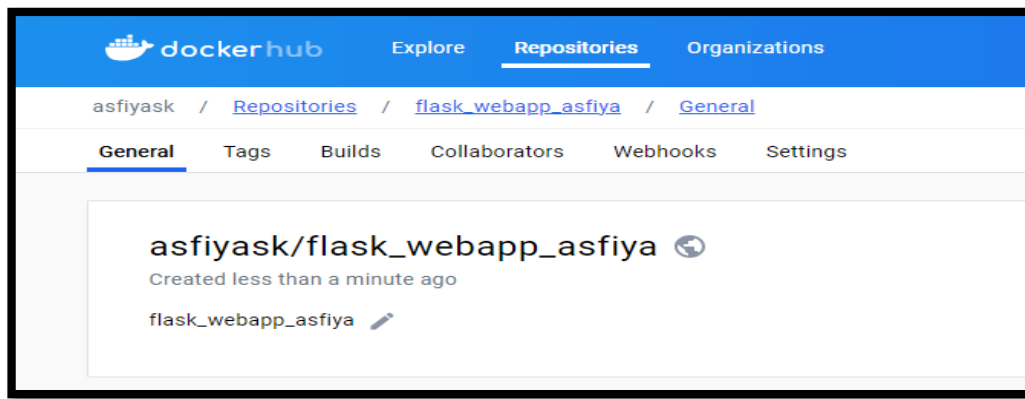
kubectl --help

kubens --help

=====

Que 8 →

- Create 1 Public Docker Hub registry named flask_webapp_yourname.



-
- Clone below repository on your system.

<https://github.com/mmumshad/simple-webapp-docker.git>

```
root@DESKTOP-800G2HF:simple-webapp-docker# git branch
```

```
* master
```

- Initialize a local repository & copy the code from above repo to your local repository in your working branch.

```
root@DESKTOP-800G2HF:simple-webapp-docker# mkdir my_scripts
```

```
root@DESKTOP-800G2HF:simple-webapp-docker# cd my_scripts/
```

```
root@DESKTOP-800G2HF:my_scripts# ll
```

```
total 0
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 21:47 ./
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 21:47 ../
```

```
root@DESKTOP-800G2HF:my_scripts# cp -pr ../Dockerfile .
```

```
root@DESKTOP-800G2HF:my_scripts# cp -pr ../app.py .
```

```
root@DESKTOP-800G2HF:my_scripts# ll
```

```
total 0
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 21:47 ./
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 21 21:47 ../
```

```
-rwxrwxrwx 1 asfiya asfiya 194 Feb 21 21:40 Dockerfile*
```

```
-rwxrwxrwx 1 asfiya asfiya 229 Feb 21 21:40 app.py*
```

-
- Once code is copied to the local repository, build the docker image & add meaningful tags with version 1 and push to Docker Hub registry.

```
root@DESKTOP-800G2HF:my_scripts# docker image build -t
asfiyask/flask_webapp_asfiya:version-1 .
```

```
[+] Building 99.5s (10/10) FINISHED
docker:default
```

```
=> [internal] load build definition from Dockerfile
0.1s
```

```
=> => transferring dockerfile: 233B
0.0s
```

```
=> [internal] load metadata for docker.io/library/ubuntu:20.04
3.2s
```

```
=> [auth] library/ubuntu:pull token for registry-1.docker.io
0.0s
```

```
=> [internal] load .dockerignore
0.0s
```

```
=> => transferring context: 2B
0.0s
```

```
=> [1/4] FROM
docker.io/library/ubuntu:20.04@sha256:bb1c41682308d7040f74d103022816d41c50d7b
0c89e9d706a74b4e548636e54 7.6s
```

```
=> => resolve
docker.io/library/ubuntu:20.04@sha256:bb1c41682308d7040f74d103022816d41c50d7b
0c89e9d706a74b4e548636e54 0.0s
```

```
=> =>
sha256:bb1c41682308d7040f74d103022816d41c50d7b0c89e9d706a74b4e548636e54
1.13kB / 1.13kB 0.0s
```

```
=> =>
sha256:a4fab1802f08df089c4b2e0a1c8f1a06f573bd1775687d07fef4076d3a2e4900 424B
/ 424B 0.0s
```

```
=> =>
sha256:18ca3f4297e795532c0d053ba443d392d5d316ee83ddee0de27f1e742a7db273
2.30kB / 2.30kB 0.0s
```

```
=> =>
sha256:8ee0874247356ecb5ea92128219660506b139dcb6cc45dcab84d98b3c6485061
27.51MB / 27.51MB 4.7s
```



```
=> => extracting
sha256:8ee0874247356ecb5ea92128219660506b139dcb6cc45dcab84d98b3c6485061
2.5s

=> [internal] load build context
0.1s

=> => transferring context: 264B
0.0s

=> [2/4] RUN apt-get update && apt-get install -y python3 python3-pip
79.9s

=> [3/4] RUN pip install flask
5.1s

=> [4/4] COPY app.py /opt/
0.1s

=> exporting to image
3.5s

=> => exporting layers
3.5s

=> => writing image
sha256:7aa75cdfd0e6a5402c2bc4d01b5db99023b6d27f2d87eee0058667379db1dce6
0.0s

=> => naming to docker.io/asfiyask/flask_webapp_asfiya:version-1
0.0s
```

```
root@DESKTOP-800G2HF:my_scripts# docker image push
asfiyask/flask_webapp_asfiya:version-1
```

```
The push refers to repository
[docker.io/asfiyask/flask_webapp_asfiya]
```

```
1e8c18d52fdf: Pushed
```

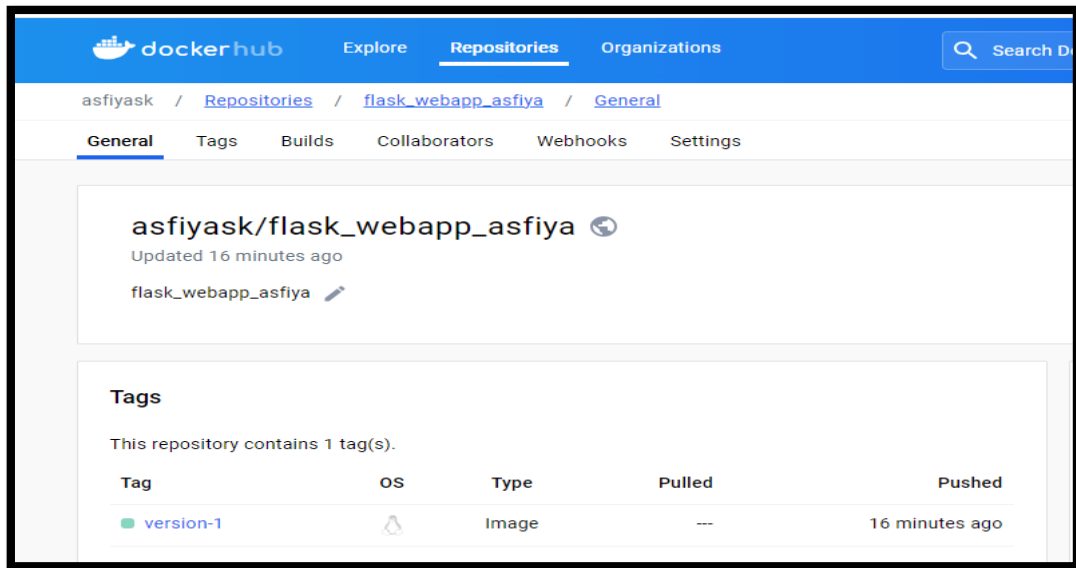
```
6153008a8793: Pushed
```

```
4af3dea163d8: Pushed
```

```
28da0445c449: Mounted from library/ubuntu
```

```
version-1: digest:
```

```
sha256:e3aa54a2c1137486ea51949890fd52232d52924bf677f05f6a7e6aca8
688589e size: 1160
```



-
- Once Images are pushed to Docker hub registries, create a directory named kube. Inside the kube directory create `deployment.yaml` file with 3 replication , labels `app: flask-webapp` , `containerPort: 8080` and add the image that you have pushed in Docker Hub registry.

```
root@DESKTOP-800G2HF:my_scripts# cd kube/
```

```
root@DESKTOP-800G2HF:kube# touch deployment.yaml
```

```
# flask-webapp-deployment yaml file
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask-webapp-deployment
  labels:
    app: flask-webapp
spec:
  replicas: 3
  selector:
    matchLabels:
      app: flask-webapp
  template:
    metadata:
      labels:
        app: flask-webapp
    spec:
      containers:
        - name: nginx
          image: asfiyask/flask_webapp_asfiya:version-1
          ports:
            - containerPort: 8080
```

```
root@DESKTOP-800G2HF:kube# touch service.yaml
```

```
apiVersion: v1
kind: Service
metadata:
  name: flask-webapp-service
spec:
  selector:
    app: flask-webapp
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
      nodePort: 30011
  type: NodePort
```

● Once a service is created try accessing the web page in the browser as below. (30011 is nodeport mentioned in service.yaml). Meanwhile open app.py from your code to understand paths & output.

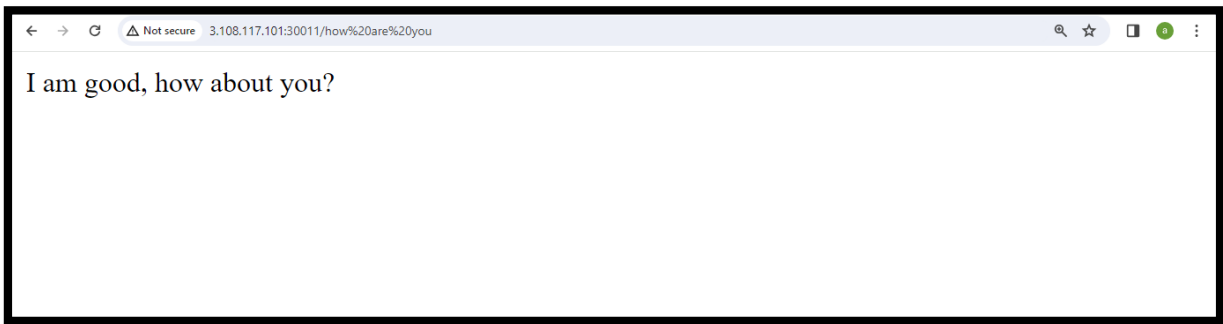
```
http://master_ip:30011/
```

```
http://master_ip:30011/how are you
```

```
root@DESKTOP-800G2HF:kube# cd
../../../../../../../../k8sCluster_kubeadm_terraform/
```

```
root@DESKTOP-800G2HF:k8sCluster_kubeadm_terraform# terraform
apply -auto-approve
```





-
- Now , update the app.py from your code and add below route above `if __name__ == "__main__"` line `@app.route('/Who are you')`
`def cloudethix(): return 'Yes, I am cloudethix, and You !!!'`

```
root@DESKTOP-800G2HF:my_scripts# cat app.py
```

```
import os
```

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route("/")
```

```
def main():
```

```
    return "Welcome!"
```

```
@app.route('/how are you')
```

```
def hello():
```

```
    return 'I am good, how about you?'
```

```
@app.route('/Who are you')
```

```
def cloudethix():
```

```
    return 'Yes, I am cloudethix, and You !!!'
```

```
if __name__ == "__main__":
```

```
    app.run()
```

- Once the file is updated , rebuild the docker image & add meaningful tags with version 2 and push to Docker Hub registry.

- Now we have the latest docker image in repo, It's time to roll out a new image. Roll out the new Image with all three ways one by one.

1. With kubectl set command
2. With kubectl edit deployment
3. With deployment.yaml file modification

```
root@DESKTOP-800G2HF:kube# k set image deployment flask-webapp-  
deployment flask-webapp-cont=asfiyask/flask_webapp_asfiya:version-1 --  
record
```

```
deployment.apps/flask-webapp-deployment image updated
```

```
root@DESKTOP-800G2HF:kube# k edit deployment flask-webapp-deployment  
deployment.apps/flask-webapp-deployment edited
```

```
root@DESKTOP-800G2HF:kube# k apply -f .
```

```
deployment.apps/flask-webapp-deployment configured
```

```
# flask-webapp-deployment yaml file  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: flask-webapp-deployment  
  labels:  
    app: flask-webapp  
spec:  
  replicas: 3  
  selector:  
    matchLabels:  
      app: flask-webapp  
  template:  
    metadata:  
      labels:  
        app: flask-webapp  
    spec:  
      containers:  
        - name: flask-webapp-cont  
          image: asfiyask/flask_webapp_asfiya:version2
```

```
ports:
  - containerPort: 8080
```

```
root@DESKTOP-800G2HF:kube# kgp
```

NAME AGE	READY	STATUS	RESTARTS
flask-webapp-deployment-78cbfb7d68-fkwtf 4m51s	1/1	Running	0
flask-webapp-deployment-78cbfb7d68-nqc4n 4m48s	1/1	Running	0
flask-webapp-deployment-78cbfb7d68-q4vvx 4m50s	1/1	Running	0

- Run the # kubectl rollout command to check status and history.

```
root@DESKTOP-800G2HF:kube# kubectl rollout status deployment flask-  
webapp-deployment
```

```
deployment "flask-webapp-deployment" successfully rolled out
```

```
root@DESKTOP-800G2HF:kube# kubectl rollout history deployment flask-  
webapp-deployment
```

```
deployment.apps/flask-webapp-deployment
```

```
REVISION  CHANGE-CAUSE
```

```
3          kubectl set image deployment flask-webapp-deployment flask-  
webapp-cont=asfiyask/flask_webapp_asfiya:version-1 --record=true
```

```
4          kubectl set image deployment flask-webapp-deployment flask-  
webapp-cont=asfiyask/flask_webapp_asfiya:version-1 --record=true
```

- Note:- Once above step 1 is done , run # kubectl rollout undo deployment command to rollback the change and then try a second way of rollout.

```
root@DESKTOP-800G2HF:kube# kubectl rollout undo deployment flask-  
webapp-deployment --to-revision=4
```

```
deployment.apps/flask-webapp-deployment rolled back
```

```
root@DESKTOP-800G2HF:kube# kubectl rollout history deployment flask-  
webapp-deployment
```

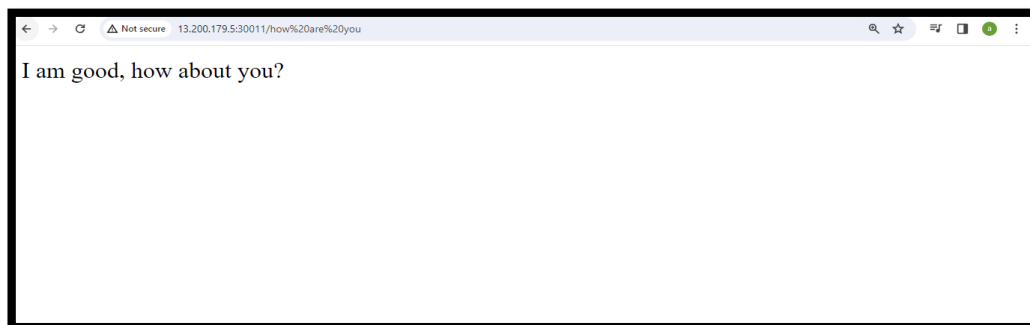
deployment.apps/flask-webapp-deployment

REVISION CHANGE-CAUSE

```
5          kubectl set image deployment flask-webapp-deployment flask-  
webapp-cont=asfiyask/flask_webapp_asfiya:version-1 --record=true
```

```
6          kubectl set image deployment flask-webapp-deployment flask-  
webapp-cont=asfiyask/flask_webapp_asfiya:version-1 --record=true
```

-
- In the browser run all three routes & notice the changes.
http://master_ip:30011/ http://master_ip:30011/how are you
http://master_ip:30011/Who are you

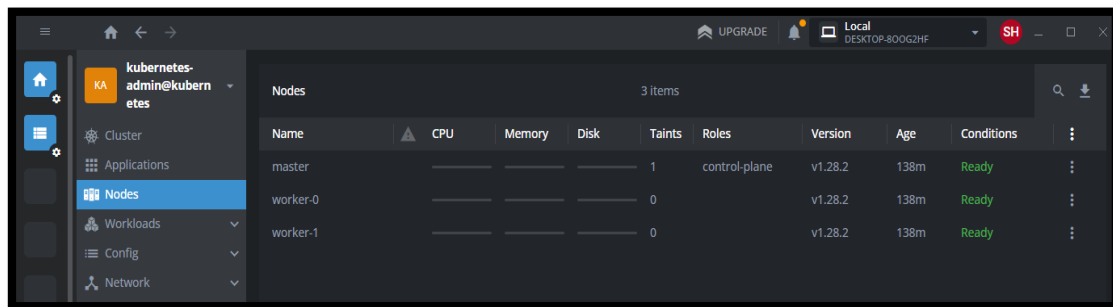


Que 9 →

- Download and install Lens & access your k8s cluster from Lens.

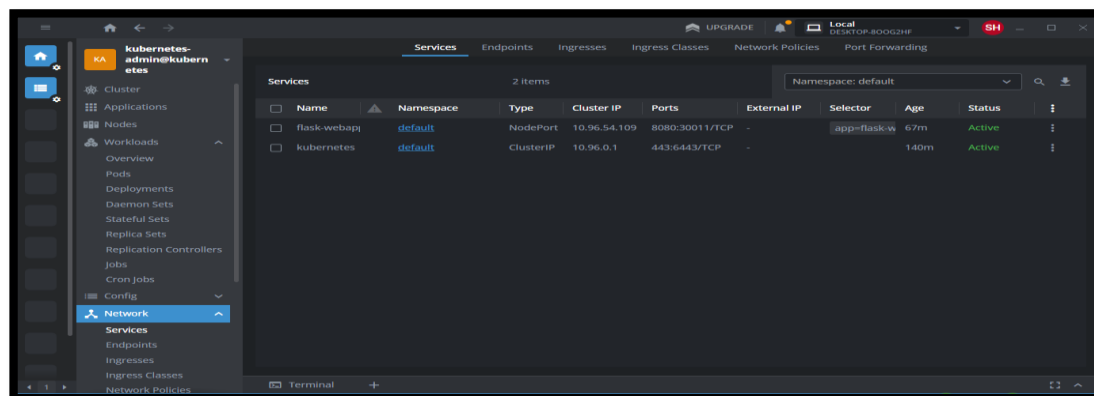
- Create nginx Pod and Nodeport service. Check the Pod logs from Lens.
- Check the service from lens. Also login to the pod shell using the lens.
- Take snaps and delete the resources that you have just created.

NODES



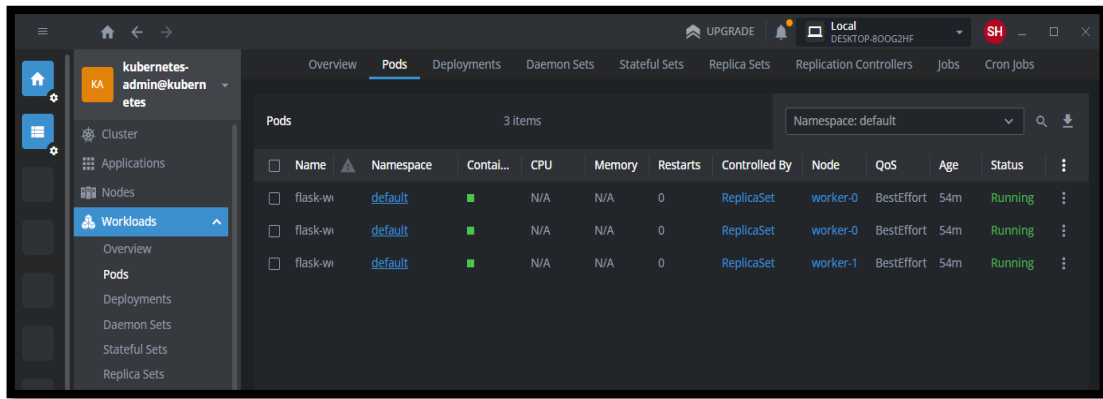
Name	CPU	Memory	Disk	Taints	Roles	Version	Age	Conditions
master				1	control-plane	v1.28.2	138m	Ready
worker-0				0		v1.28.2	138m	Ready
worker-1				0		v1.28.2	138m	Ready

SERVICES

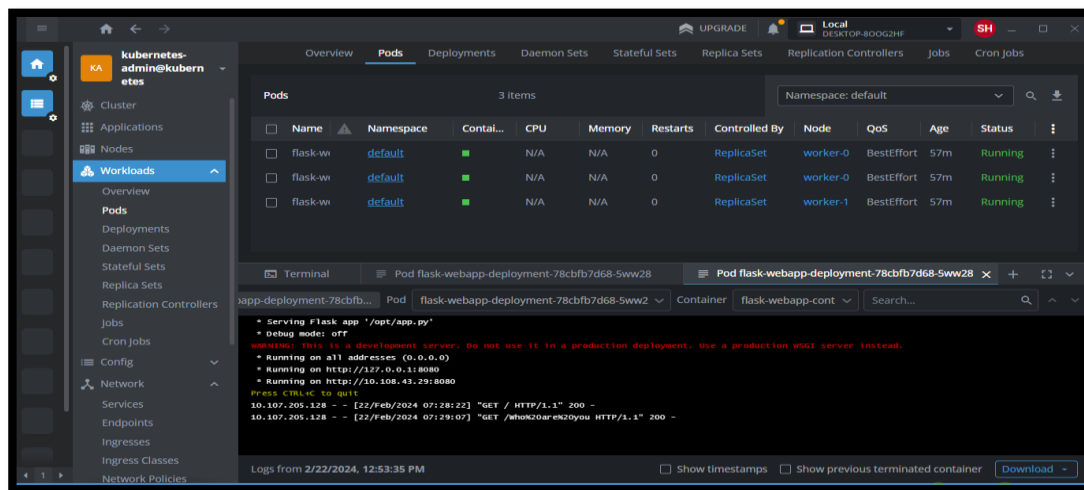


Name	Namespace	Type	Cluster IP	Ports	External IP	Selector	Age	Status
flask-webapi	default	NodePort	10.96.54.109	8080:30011/TCP	-	app=flask-w	67m	Active
kubernetes	default	ClusterIP	10.96.0.1	443:6443/TCP	-		140m	Active

PODS

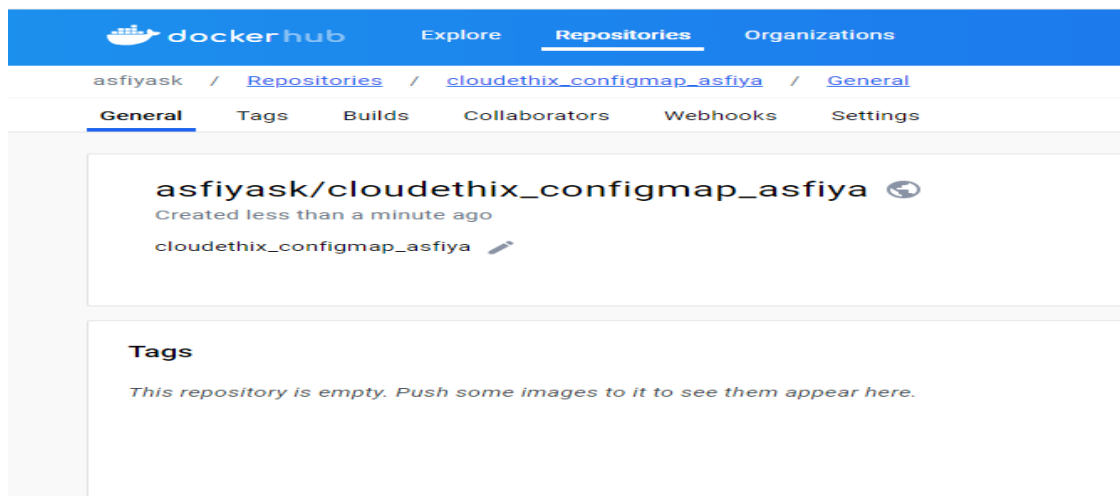


POD LOGS



Que 10 →

- Create 1 Public Docker Hub registry named `cloudethix_configmap_yourname`.



- Clone below repository on your system.
<https://github.com/zembutsu/docker-sample-nginx.git>

```
root@DESKTOP-800G2HF:configmap-volume# git clone
git@github.com:zembutsu/docker-sample-nginx.git
Cloning into 'docker-sample-nginx'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 22 (delta 7), reused 6 (delta 6), pack-reused 10
Receiving objects: 100% (22/22), done.
Resolving deltas: 100% (7/7), done.
```

```
root@DESKTOP-800G2HF:configmap-volume# ll
total 0
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 14:01 ./
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 14:00 ../
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 14:01 docker-sample-nginx/
root@DESKTOP-800G2HF:configmap-volume# cd docker-sample-nginx/
root@DESKTOP-800G2HF:docker-sample-nginx# ll
total 4
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 14:01 ./
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 14:01 ../
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 14:01 .git/
-rwxrwxrwx 1 asfiya asfiya 95 Feb 22 14:01 Dockerfile*
-rwxrwxrwx 1 asfiya asfiya 1084 Feb 22 14:01 LICENSE*
-rwxrwxrwx 1 asfiya asfiya 73 Feb 22 14:01 README.md*
-rwxrwxrwx 1 asfiya asfiya 286 Feb 22 14:01 default.conf*
-rwxrwxrwx 1 asfiya asfiya 103 Feb 22 14:01 index.html*
```

- Initialize a local repository & copy the code from above repo to your local repository in the working branch.

- Once code is copied , build a docker image from docker file and add meaningful tags and push to docker hub repository.

```
root@DESKTOP-800G2HF:LocalProjectHub# docker image build -t asfiyask/cloudethix_configmap_asfiya:v1 .
[*] Building 1.0s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 132B
=> [internal] load metadata for docker.io/library/nginx:alpine
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/3] FROM docker.io/library/nginx:alpine@sha256:6a2f8b28e45c4adea04ec207a251fd4a2df03ddc930f782af51e315ebc76e9a9
=> => resolve docker.io/library/nginx:alpine@sha256:6a2f8b28e45c4adea04ec207a251fd4a2df03ddc930f782af51e315ebc76e9a9
=> [internal] load build context
=> => transferring context: 469B
=> CACHED [2/3] COPY default.conf /etc/nginx/conf.d/
=> CACHED [3/3] COPY index.html /usr/share/nginx/html/
```

```
root@DESKTOP-800G2HF:docker-sample-nginx# docker image push
asfiyask/cloudethix_configmap_asfiya:v1
```

The push refers to repository
[docker.io/asfiyask/cloudethix_configmap_asfiya]

788ad69e55bb: Pushed

b397c7fc2d49: Mounted from asfiyask/cloudethix_release_nginx_asfiya

667a247707f0: Mounted from asfiyask/cloudethix_release_nginx_asfiya

d8527026595f: Mounted from asfiyask/cloudethix_release_nginx_asfiya

2593b08e5428: Mounted from asfiyask/cloudethix_release_nginx_asfiya

9909978d630d: Mounted from asfiyask/cloudethix_release_nginx_asfiya

c5140fc719dd: Mounted from asfiyask/cloudethix_release_nginx_asfiya

3137f8f0c641: Mounted from asfiyask/cloudethix_release_nginx_asfiya

718db50a47c0: Mounted from asfiyask/cloudethix_release_nginx_asfiya

aedc3bda2944: Mounted from asfiyask/cloudethix_release_nginx_asfiya

v1: digest:

sha256:b49f72a7a19241426cd108efada55296d9bc1e14c8232a8573a15ead9f9b1b5f size: 2403

-
- Once Images are pushed to Docker hub registries, create a directory named kube. Inside the kube directory create deployment.yaml file with 3 replication , labels app: frontend-webapp , containerPort: 80 and add the image that you have pushed in Docker Hub registry.

```
root@DESKTOP-800G2HF:docker-sample-nginx# mkdir kube
```

```
root@DESKTOP-800G2HF:docker-sample-nginx# cd kube/
```

```
root@DESKTOP-800G2HF:kube# touch deployment.yaml
```

```
# flask-webapp-deployment yaml file
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend-webapp-deployment
  labels:
    app: frontend-webapp
spec:
  replicas: 3
  selector:
    matchLabels:
      app: frontend-webapp
  template:
    metadata:
      labels:
        app: frontend-webapp
    spec:
      containers:
        - name: frontend-webapp-cont
          image: asfiyask/cloudethix_configmap_asfiya:v1
          ports:
            - containerPort: 80
```

-
- Create one service.yaml file with type nodeport & select frontend-webapp pod with port 80 & targetPort 80 with any nodePort between range 30000-32768.

```
root@DESKTOP-800G2HF:kube# touch service.yaml
```

```
apiVersion: v1
kind: Service
metadata:
  name: frontend-webapp-service
spec:
  selector:
    app: frontend-webapp

  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

```
nodePort: 30012
type: NodePort
```

```
root@DESKTOP-800G2HF:kube# k apply -f .
deployment.apps/frontend-webapp-deployment created
service/frontend-webapp-service unchanged
```

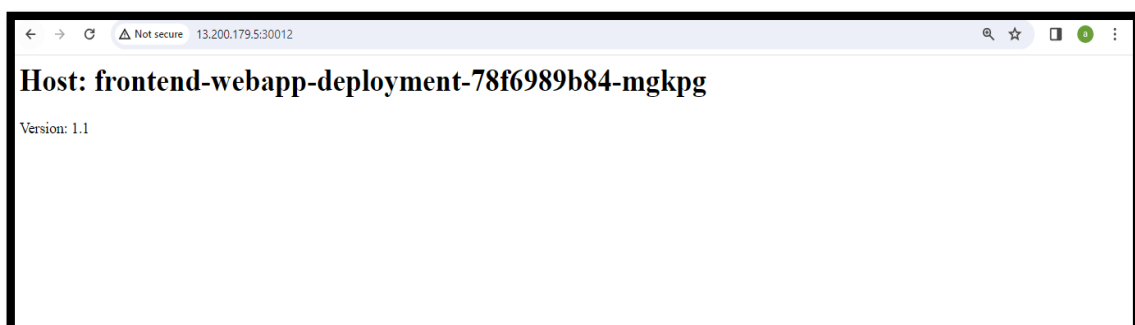
```
root@DESKTOP-800G2HF:kube# kgp
```

NAME	READY	STATUS
frontend-webapp-deployment-78f6989b84-h8lgw	1/1	Running
0		
10s		
frontend-webapp-deployment-78f6989b84-mgkpg	1/1	Running
0		
10s		
frontend-webapp-deployment-78f6989b84-p67br	1/1	Running
0		
10s		

```
root@DESKTOP-800G2HF:kube# kgs
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP
frontend-webapp-service	NodePort	10.100.69.39	<none>
80:30012/TCP			
41s			
kubernetes	ClusterIP	10.96.0.1	<none>
443/TCP			
3h4m			

-
- Once the service is created try accessing the web page in the browser as below. Notice the changes & take the snap.



- Now create a configmap.yaml file with below data & delete the deployment

that you have created.

```
<html>
<body>
<h1> I am Cloudethix Team, Are you ?!! </h1>
Version: 1.1
</body>
</html>
```

```
root@DESKTOP-800G2HF:kube# k delete -f .
deployment.apps "frontend-webapp-deployment" deleted
service "frontend-webapp-service" deleted
```

```
root@DESKTOP-800G2HF:kube# kgp
```

NAME	READY	STATUS
RESTARTS AGE		
flask-webapp-deployment-78cbfb7d68-5ww28 106m	1/1	Running 0
flask-webapp-deployment-78cbfb7d68-mlzkk 106m	1/1	Running 0
flask-webapp-deployment-78cbfb7d68-z5pt4 106m	1/1	Running 0

```
root@DESKTOP-800G2HF:kube# touch configmap.yaml
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-configmap
data:
  index.html: |
    <html>
    <body>
    <h1> I am Cloudethix Team, Are you ?!! </h1>
    Version: 1.1
    </body>
```

</html>

-
- Then update the same deployment.yaml file and mount configmap as volume on container using volumeMounts with mountPath /usr/share/nginx/html/

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend-webapp-deployment
  labels:
    app: frontend-webapp
spec:
  replicas: 3
  selector:
    matchLabels:
      app: frontend-webapp
  template:
    metadata:
      labels:
        app: frontend-webapp
    spec:
      containers:
        - name: frontend-webapp-cont
          image: asfiyask/cloudethix_configmap_asfiya:v1
          ports:
            - containerPort: 80
          volumeMounts:
            - name: config-volume
              mountPath: /usr/share/nginx/html
      volumes:
        - name: config-volume
          configMap:
            name: my-configmap
```

-
- Now it's time to create configmap & deployment. Once created , try to access the webpage in the browser & confirm that the index page is the same as we have in configmap.

```
root@DESKTOP-800G2HF:kube# k apply -f .
```

```
configmap/my-configmap created
```

```
deployment.apps/frontend-webapp-deployment created
```

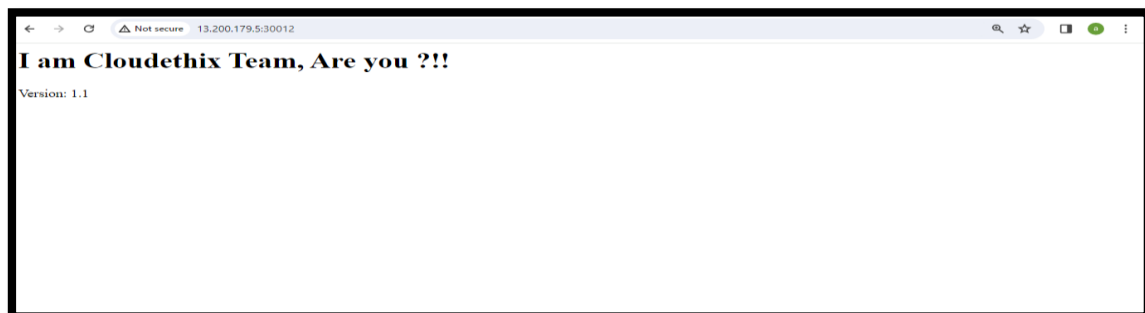
service/frontend-webapp-service created

root@DESKTOP-800G2HF:kube# kgp

NAME	READY	STATUS
RESTARTS AGE		
frontend-webapp-deployment-6b8bd8b577-5bhgv 2s	1/1	Running 0
frontend-webapp-deployment-6b8bd8b577-jvmxq 2s	1/1	Running 0
frontend-webapp-deployment-6b8bd8b577-sblmn 2s	1/1	Running 0

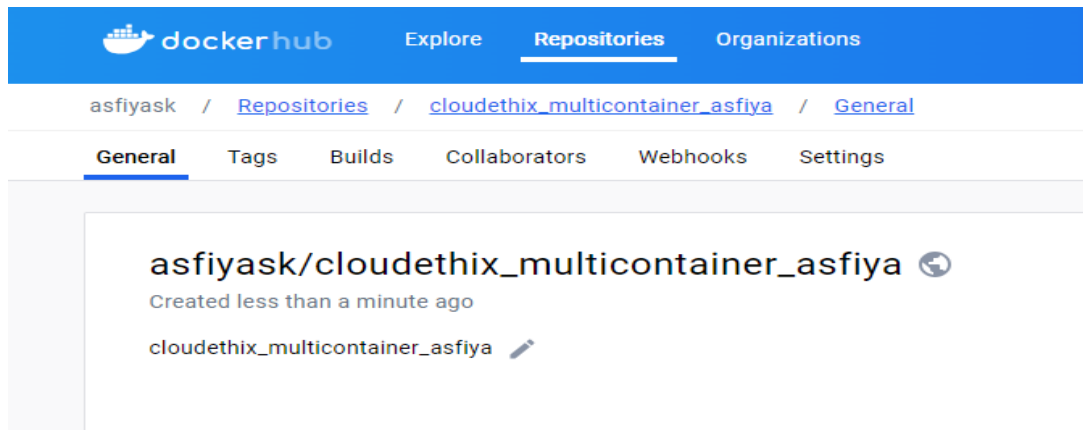
root@DESKTOP-800G2HF:kube# kgs

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP
PORT(S) AGE			
frontend-webapp-service 80:30012/TCP 12s	NodePort	10.106.20.243	<none>



Que 11 →

- Create 1 Public Docker Hub registry named cludethix_multicontainer_yourname.



- Clone below repository on your system.

<https://github.com/janakiramm/Kubernetes-multi-container-pod.git>

```
root@DESKTOP-800G2HF:my_scripts# mkdir multicontainer
```

```
root@DESKTOP-800G2HF:my_scripts# cd multicontainer/
```

```
root@DESKTOP-800G2HF:multicontainer# git clone
```

```
git@github.com:janakiramm/Kubernetes-multi-container-pod.git
```

```
Cloning into 'Kubernetes-multi-container-pod'...
```

```
remote: Enumerating objects: 51, done.
```

```
remote: Total 51 (delta 0), reused 0 (delta 0), pack-reused 51
```

```
Receiving objects: 100% (51/51), 88.14 KiB | 308.00 KiB/s, done.
```

```
Resolving deltas: 100% (21/21), done.
```

```
root@DESKTOP-800G2HF:multicontainer# ll
```

```
total 0
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 15:03 ./
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 15:02 ../
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 15:03 Kubernetes-multi-  
container-pod/
```

```
root@DESKTOP-800G2HF:multicontainer# cd Kubernetes-multi-container-  
pod/
```

```
root@DESKTOP-800G2HF:multicontainer# cd Kubernetes-multi-container-  
pod/
```

```
root@DESKTOP-800G2HF:Kubernetes-multi-container-pod# ll
```

```
total 120
```

```
drwxrwxrwx 1 asfiya asfiya    512 Feb 22 15:03 ./
drwxrwxrwx 1 asfiya asfiya    512 Feb 22 15:03 ../
drwxrwxrwx 1 asfiya asfiya    512 Feb 22 15:03 .git/
-rwxrwxrwx 1 asfiya asfiya      9 Feb 22 15:03 .gitignore*
drwxrwxrwx 1 asfiya asfiya    512 Feb 22 15:03 Build/
drwxrwxrwx 1 asfiya asfiya    512 Feb 22 15:03 Deploy/
-rwxrwxrwx 1 asfiya asfiya  2550 Feb 22 15:03 README.md*
-rwxrwxrwx 1 asfiya asfiya 116003 Feb 22 15:03 multi-container-
pod.png*
```

- Once code is copied , go to the Build directory and build docker image from docker file and add meaningful tags and push to docker hub repository.

```
root@DESKTOP-800G2HF:Kubernetes-multi-container-pod# cd Build/
```

```
root@DESKTOP-800G2HF:Build# ll
```

```
total 4
```

```
drwxrwxrwx 1 asfiya asfiya    512 Feb 22 15:03 ./
drwxrwxrwx 1 asfiya asfiya    512 Feb 22 15:03 ../
-rwxrwxrwx 1 asfiya asfiya     62 Feb 22 15:03 Dockerfile*
-rwxrwxrwx 1 asfiya asfiya  1607 Feb 22 15:03 app.py*
-rwxrwxrwx 1 asfiya asfiya   242 Feb 22 15:03 docker-compose.yml*
-rwxrwxrwx 1 asfiya asfiya    24 Feb 22 15:03 requirements.txt*
```

```
root@DESKTOP-800G2HF:Build# docker image build -t asfiyask/cloudethix_multicontainer_asfiya:va . --load
[+] Building 63.2s (11/11) FINISHED                                docker-container:strange_perlman
=> [internal] load build definition from Dockerfile                0.1s
=> => transferring dockerfile: 99B                                0.0s
=> [internal] load metadata for docker.io/library/python:2.7-onbuild 12.4s
=> [auth] library/python:pull token for registry-1.docker.io       0.0s
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                       0.0s
=> [1/1] FROM docker.io/library/python:2.7-onbuild@sha256:5af88e1d011bf7e845e83813712d9f91bela39e2ede092008fc53e0a8ce1333b 0.1s
=> => resolve docker.io/library/python:2.7-onbuild@sha256:5af88e1d011bf7e845e83813712d9f91bela39e2ede092008fc53e0a8ce1333b 0.1s
=> [internal] load build context                                  0.1s
=> => transferring context: 130B                                       0.0s
=> CACHED [2/1] COPY requirements.txt /usr/src/app/                0.0s
=> CACHED [3/1] RUN pip install --no-cache-dir -r requirements.txt 0.0s
=> CACHED [4/1] COPY . /usr/src/app                               0.0s
=> exporting to docker image format                               50.3s
=> => exporting layers                                                0.9s
=> => exporting manifest sha256:61505b70c29b38c7a18c1020f764121e62276f309621fa00341f235def6fb2b1 0.1s
=> => exporting config sha256:4f5520807740169db87462a9b7aled0f551d1470433546554067a64110327aa9 0.1s
=> => sending tarball                                                49.3s
=> importing to docker                                           39.3s
root@DESKTOP-800G2HF:Build# docker images
REPOSITORY                                TAG                IMAGE ID            CREATED            SIZE
asfiyask/cloudethix_multicontainer_asfiya  va                4f5520807740      2 minutes ago     691MB
```

```
root@DESKTOP-800G2HF:Build# docker image push
asfiyask/cloudethix_multicontainer_asfiya:va
```

The push refers to repository
[docker.io/asfiyask/cloudethix_multicontainer_asfiya]

8f18896fd729: Pushed

b5fc42bbcb37: Pushed

d0fb1324b33b: Pushed

3e397f5b8357: Pushed

e257add70b4b: Pushed

ce7e990ce056: Pushed

633d23790c1d: Pushed

d071a18d9802: Pushed

8451f9fe0016: Pushed

858cd8541f7e: Pushed

a42d312a03bb: Pushed

dd1eb1fd7e08: Pushed

va: digest:

sha256:1f90ebab5e9f5c41f018558c801f88eae6aa150c83653467e4e04ab62103d5a
5 size: 2844

-
- Now go to the deploy directory and notice the files.
 - Here, web-pod-1.yml file will create the pod with two containers (Multi container). Take a note of labels, name of containers and ports. Also, please make sure you will update the python container image that you have pushed to your docker registry.

```
apiVersion: "v1"
kind: Pod
metadata:
  name: web1
  labels:
    name: web
    app: demo
spec:
  containers:
    - name: redis
      image: redis
      ports:
        - containerPort: 6379
```

```

        name: redis
        protocol: TCP
-   name: python
    image: asfiyask/cloudethix_multicontainer_asfiya:va
    env:
      - name: "REDIS_HOST"
        value: "localhost"
    ports:
      - containerPort: 5000
        name: http
        protocol: TCP

```

- Now, open web-svc.yml file and notice service Type , selectors & targetPort. Apply the file

```

#web-svc.yml
apiVersion: v1
kind: Service
metadata:
  name: web
  labels:
    name: web
    app: demo
spec:
  selector:
    name: web
  type: NodePort
  ports:
    - port: 80
      name: http
      targetPort: 5000
      protocol: TCP

```

- Now open db-pod.yml & notice the labels , name , Image, containerPort and apply the file.

```

apiVersion: "v1"
kind: Pod
metadata:
  name: mysql
  labels:
    name: mysql
    app: demo
spec:
  containers:
    - name: mysql
      image: mysql:5.7.25
      ports:

```

```

- containerPort: 3306
  protocol: TCP
env:
-
  name: "MYSQL_ROOT_PASSWORD"
  value: "password"

```

- Now open the db-svc.yml file and notice service Type , selectors & targetPort. Apply the file.

```

apiVersion: v1
kind: Service
metadata:
  name: mysql
  labels:
    name: mysql
    app: demo
spec:
  ports:
  - port: 3306
    name: mysql
    targetPort: 3306
  selector:
    name: mysql
    app: demo

```

-
- Now , from the command line run below urls & notice the changes. # curl http://\$NODE_IP:\$NODE_PORT/init Initialize the database with sample schema

```
root@DESKTOP-800G2HF:Deploy# curl http://13.200.179.5:31424/init
```

```
DB Init doneroot@DESKTOP-800G
```

- Now it's time to Insert some sample data. Make sure you will use correct \$NODE_IP:\$NODE_PORT

```
# curl -i -H "Content-Type: application/json" -X POST -d
'{"uid": "1",
"user":"John Doe"}' http://$NODE_IP:$NODE_PORT/users/add
```

```
# curl -i -H "Content-Type: application/json" -X POST -d
'{"uid": "2",
```

```
"user":"Jane Doe"}' http://$NODE_IP:$NODE_PORT/users/add
```

```
{# curl -i -H "Content-Type: application/json" -X POST -d  
'{"uid": "3",
```

```
"user":"Bill Colls"}' http://$NODE_IP:$NODE_PORT/users/add
```

```
# curl -i -H "Content-Type: application/json" -X POST -d  
'{"uid": "4",
```

```
"user":"Mike Taylor"}' http://$NODE_IP:$NODE_PORT/users/add
```

```
@DESKTOP-800Gcurl -i -H "Content-Type: application/json" -X POST -d  
'{"uid": "1",{"uid": "1",
```

```
"user":"John Doe"}' http://13.200.179.5:31424/users/add
```

```
HTTP/1.0 200 OK
```

```
Content-Type: application/json
```

```
Content-Length: 5
```

```
Server: Werkzeug/1.0.1 Python/2.7.15
```

```
Date: Thu, 22 Feb 2024 11:29:12 GMT
```

```
root@DESKTOP-800G2HF:Deploy# curl -i -H "Content-Type:  
application/json" -X POST -d '{"uid": "2",
```

```
"user":"Jane Doe"}' http://13.200.179.5:31424/users/add
```

```
HTTP/1.0 200 OK
```

```
Content-Type: application/json
```

```
Content-Length: 5
```

```
Server: Werkzeug/1.0.1 Python/2.7.15
```

```
Date: Thu, 22 Feb 2024 11:40:35 GMT
```

```
root@DESKTOP-800G2HF:Deploy# curl -i -H "Content-Type:  
application/json" -X POST -d '{"uid": "3",
```

```
"user":"Bill Colls"}' http://13.200.179.5:31424/users/add
```

```
HTTP/1.0 200 OK
```

```
Content-Type: application/json
```

Content-Length: 5

Server: Werkzeug/1.0.1 Python/2.7.15

Date: Thu, 22 Feb 2024 11:41:27 GMT

```
root@DESKTOP-800G2HF:Deploy# curl -i -H "Content-Type:
application/json" -X POST -d '{"uid": "4",
"user":"Mike Taylor"}' http://13.200.179.5:31424/users/add
HTTP/1.0 200 OK
```

Content-Type: application/json

Content-Length: 5

Server: Werkzeug/1.0.1 Python/2.7.15

Date: Thu, 22 Feb 2024 11:34:46 GMT

● The second time you access the data, it appends '(c)' indicating that it is pulled from the Redis cache.

```
root@DESKTOP-800G2HF:Deploy# curl http://13.200.179.5:31424/users/1
John Doe
```

● Also, try to access mysql shell i.e db pod & run select * from the users table. check app.py for DB related information.

```
root@DESKTOP-800G2HF:Deploy# systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2024-02-22 17:32:30 IST; 7s ago
     Process: 391761 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 391780 (mysqld)
      Status: "Server is operational"
        Tasks: 38 (limit: 4639)
       Memory: 365.5M
        CGroup: /system.slice/mysql.service
                └─391780 /usr/sbin/mysqld
```

```
root@DESKTOP-800G2HF:Deploy# mysql -u root -p
```

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 8

Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
root@DESKTOP-800G2HF:Deploy# mysql -u root -p
```

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 9

Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> SHOW DATABASES;
```

```
+-----+
| Database          |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+-----+
```

```
4 rows in set (0.01 sec)
```



```
mysql> USE mysql;
```

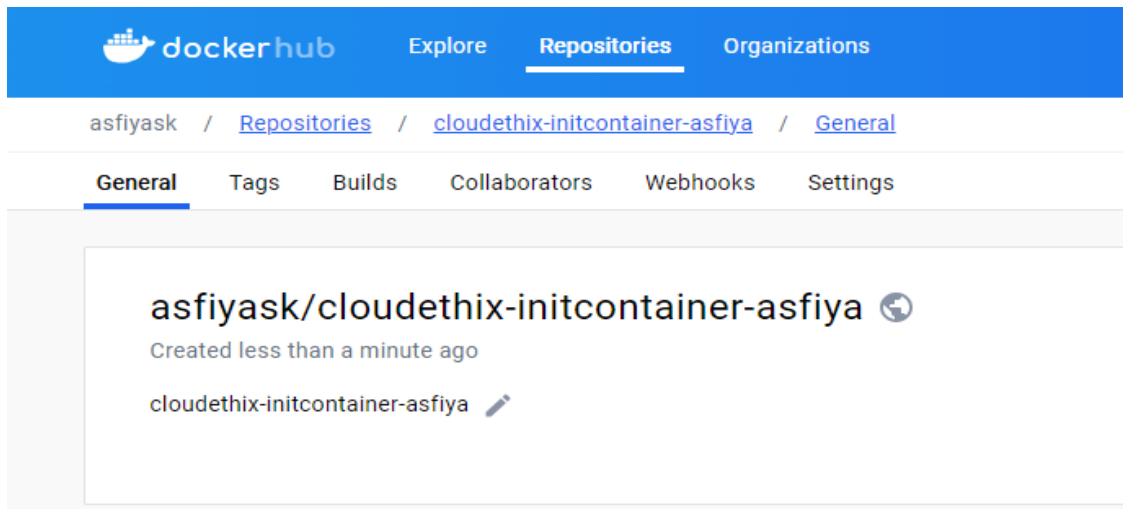
Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

```
mysql> SHOW TABLES;
```

+-----+	
Tables_in_mysql	
+-----+	
columns_priv	
component	
db	
default_roles	
engine_cost	
func	
general_log	
global_grants	
gtid_executed	
help_category	
help_keyword	
help_relation	
help_topic	
innodb_index_stats	
innodb_table_stats	
password_history	
plugin	
procs_priv	
proxies_priv	
replication_asynchronous_connection_failover	
replication_asynchronous_connection_failover_managed	
replication_group_configuration_version	
replication_group_member_actions	



-
- Clone below repository on your system.

<https://github.com/janakiramm/simpleapp.git>

```
root@DESKTOP-800G2HF:my_scripts# mkdir Initcontainer
```

```
root@DESKTOP-800G2HF:my_scripts# cd Initcontainer/
```

```
root@DESKTOP-800G2HF:Initcontainer# git clone
git@github.com:janakiramm/simpleapp.git
```

```
Cloning into 'simpleapp'...
```

```
remote: Enumerating objects: 47, done.
```

```
remote: Total 47 (delta 0), reused 0 (delta 0), pack-reused 47
```

```
Receiving objects: 100% (47/47), 8.20 KiB | 147.00 KiB/s, done.
```

```
Resolving deltas: 100% (9/9), done.
```

```
root@DESKTOP-800G2HF:Initcontainer# ll
```

```
total 0
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 17:52 ./
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 17:50 ../
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 17:52 simpleapp/
```

```
root@DESKTOP-800G2HF:Initcontainer# cd simpleapp/
```

```
root@DESKTOP-800G2HF:simpleapp# ll
```

```
total 0
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 17:52 ./
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 17:52 ../
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 17:52 .git/
-rwxrwxrwx 1 asfiya asfiya 85 Feb 22 17:52 Dockerfile*
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 17:52 html/
-rwxrwxrwx 1 asfiya asfiya 69 Feb 22 17:52 wrapper.sh*
```

- Once code is copied , go to the Build directory and build docker image from docker file and add meaningful tags and push to docker hub repository.

```
root@DESKTOP-800G2HF:simpleapp# docker image build -t asfiyask/cloudethix-initcontainer-asfiya:v1 . --load
[+] Building 0.0s (0/0)
[+] Building 3.9s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 122B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/3] FROM docker.io/library/nginx:latest@sha256:c26ae7472d624ba1fafd296e73cecc4f93f853088e6a9c13c0d52f6ca5865107
=> => resolve docker.io/library/nginx:latest@sha256:c26ae7472d624ba1fafd296e73cecc4f93f853088e6a9c13c0d52f6ca5865107
=> [internal] load build context
=> => transferring context: 90B
=> CACHED [2/3] COPY wrapper.sh /
=> CACHED [3/3] COPY html /usr/share/nginx/html
=> exporting to docker image format
=> exporting layers
=> => exporting manifest sha256:08ecf18ec19026c60e608b16ed6f0cd4baaf09d36d805ca4f362c18e3270d7e4
=> => exporting config sha256:aae376c8baa04db77ab2f1ffc50a0a202c7c9a27f45f214bae03a4f242e5af07
=> => sending tarball
=> importing to docker
```

REPOSITORY	TAG
IMAGE ID	CREATED
SIZE	
asfiyask/cloudethix-initcontainer-asfiya	v1
aae376c8baa0	21 seconds ago
187MB	

```
root@DESKTOP-800G2HF:simpleapp# docker image push asfiyask/cloudethix-initcontainer-asfiya:v1
```

The push refers to repository [docker.io/asfiyask/cloudethix-initcontainer-asfiya]

```
e51c056ac20f: Pushed
300802a9a414: Pushed
61a7fb4dabcd: Mounted from asfiyask/nginx
bcc6856722b7: Mounted from asfiyask/nginx
188d128a188c: Mounted from asfiyask/nginx
7d52a4114c36: Mounted from asfiyask/nginx
3137f8f0c641: Mounted from asfiyask/cloudethix_configmap_asfiya
84619992a45b: Mounted from asfiyask/nginx
ceb365432eec: Mounted from asfiyask/nginx
```

```
vI: digest:
sha256:e6d0ad824e66110b1f72afd8d494d2edd3a5dda9377530b60809503eb499d4d
1 size: 2192
```

- Once Images are pushed to Docker hub registries, create a directory named kube. Inside the kube directory create deployment.yaml file with 3 replication , label app: simpleapp-webapp , containerPort: 80 and add the image that you have pushed in Docker Hub registry.

```
root@DESKTOP-800G2HF:simpleapp# mkdir kube
```

```
root@DESKTOP-800G2HF:simpleapp# cd kube/
```

```
root@DESKTOP-800G2HF:kube# touch deployment.yaml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: simpleapp-webapp-deployment
  labels:
    app: simpleapp-webapp
spec:
  replicas: 3
  selector:
    matchLabels:
      app: simpleapp-webapp
  template:
    metadata:
      labels:
        app: simpleapp-webapp
    spec:
      containers:
        - name: simpleapp-webapp-cont
          image: asfiyask/cloudethix-initcontainer-asfiya:vI
          ports:
            - containerPort: 80
```

- Create one service.yaml file with type nodeport & select simpleapp-webapp pod with port 80 & targetPort 80 with any nodePort between range 30000-32768.

```
apiVersion: v1
kind: Service
metadata:
  name: simpleapp-webapp-service
spec:
  selector:
    app: simpleapp-webapp
```

```
ports:
- protocol: TCP
  port: 80
  targetPort: 80
  nodePort: 30013
type: NodePort
```

```
root@DESKTOP-800G2HF:kube# k apply -f .
```

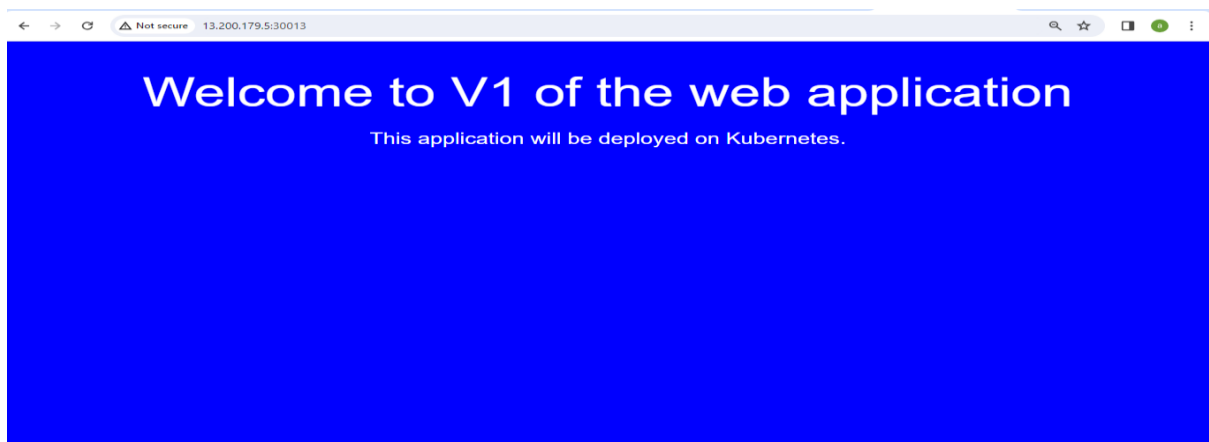
```
root@DESKTOP-800G2HF:kube# kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
simpleapp-webapp-deployment-84d879f6bc-d7hfd	1/1	Running	0	41s
simpleapp-webapp-deployment-84d879f6bc-nwjw4	1/1	Running	0	41s
simpleapp-webapp-deployment-84d879f6bc-rhk2j	1/1	Running	0	41s

```
root@DESKTOP-800G2HF:kube# kubectl get svc
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
simpleapp-webapp-service	NodePort	10.109.247.99	<none>	80:30013/TCP	70s

-
- Open the webpage in the browser and notice the changes and capture the snap.



-
- Then delete the deployment that you have just created.

```
root@DESKTOP-800G2HF:kube# k delete -f .
deployment.apps "simpleapp-webapp-deployment" deleted
service "simpleapp-webapp-service" deleted
```

- Update the deployment.yaml file and add volumeMounts with mountPath

/usr/share/nginx/html from emptyDir: {} volume.

- Once above changes are added, add initContainers block with below

parameters. Also add volumeMounts for Init Container with mountPath

"/work-dir" from emptyDir: {} volume.

initContainers:

- name: install

image: busybox:1.28

command:

- wget

- "-O"

- "/work-dir/index.html"

- http://info.cern.ch

volumeMounts:

- name: workdir

mountPath: "/work-dir"

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: simpleapp-webapp
spec:
  selector:
    matchLabels:
      app: simpleapp-webapp
  template:
    metadata:
      labels:
```

```

    app: simpleapp-webapp
spec:
  initContainers:
  - name: install
    image: busybox:1.28
    command:
    - wget
    - "-O"
    - "/work-dir/index.html"
    - "http://info.cern.ch"
    volumeMounts:
    - name: workdir
      mountPath: "/work-dir"
  containers:
  - name: simpleapp-webapp
    image: asfiyask/cloudethix-initcontainer-asfiya:initt
    ports:
    - containerPort: 80
    volumeMounts:
    - name: html-volume
      mountPath: /usr/share/nginx/html
    - name: workdir
      mountPath: /work-dir
  volumes:
  - name: html-volume
    emptyDir: {}
  - name: workdir
    emptyDir: {}

```

-
- Add volumes with emptyDir: {} in deployment.yaml file.

```

apiVersion: v1
kind: Service
metadata:
  name: simpleapp-webapp-service
spec:
  selector:
    app: simpleapp-webapp
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
    nodePort: 30016
  type: NodePort

```

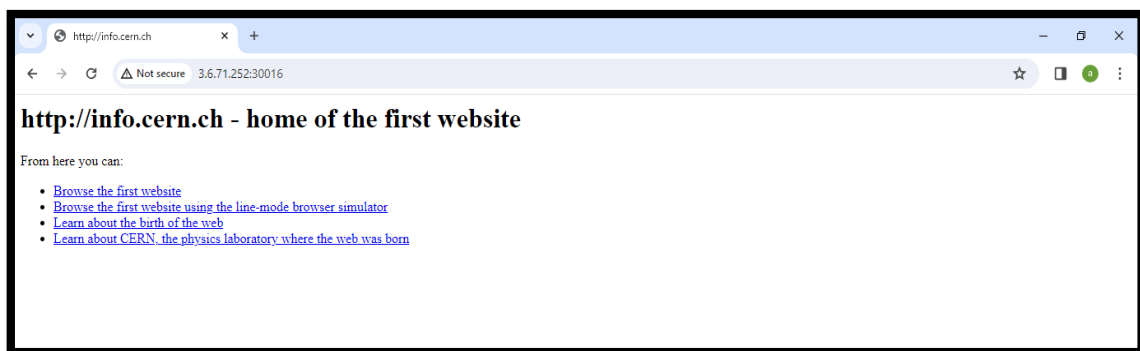

WRAPPER.SH FILE OUTPUT

```
#!/bin/bash

echo "Nginx is running..."
cp -r /work-dir/index.html /usr/share/nginx/html/
exec nginx -g "daemon off;"
```

- Once the deployment.yaml file is ready, create the deployment & access the page in the browser and notice the changes.

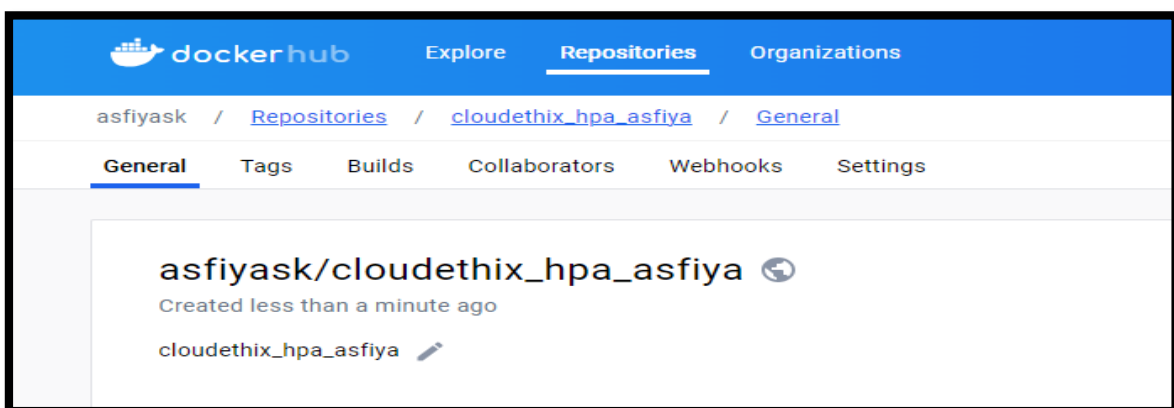
Ss



=====

Que 13 →

- Create 1 Public Docker Hub registry named cloudethix_hpa_yourname.



- Clone below repository on your system.
<https://github.com/vivekamin/kubernetes-hpa-example.git>

```
root@DESKTOP-800G2HF:my_scripts# mkdir hpa
root@DESKTOP-800G2HF:my_scripts# cd hpa/
```

```
root@DESKTOP-800G2HF:hpa# git clone
git@github.com:vivekamin/kubernetes-hpa-example.git
Cloning into 'kubernetes-hpa-example'...
remote: Enumerating objects: 26, done.
remote: Total 26 (delta 0), reused 0 (delta 0), pack-reused 26
Receiving objects: 100% (26/26), done.
Resolving deltas: 100% (9/9), done.
```

```
root@DESKTOP-800G2HF:hpa# ll
total 0
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 18:48 ./
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 18:47 ../
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 18:48 kubernetes-hpa-example/
```

```
root@DESKTOP-800G2HF:hpa# cd kubernetes-hpa-example/
```

```
root@DESKTOP-800G2HF:kubernetes-hpa-example# ll
total 4
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 18:48 ./
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 18:48 ../
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 18:48 .git/
-rwxrwxrwx 1 asfiya asfiya 127 Feb 22 18:48 Dockerfile*
-rwxrwxrwx 1 asfiya asfiya 2788 Feb 22 18:48 README.md*
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 18:48 k8s/
-rwxrwxrwx 1 asfiya asfiya 272 Feb 22 18:48 package.json*
drwxrwxrwx 1 asfiya asfiya 512 Feb 22 18:48 src/
```

- Once code is copied , build a docker image from the docker file and add meaningful tags and push to the docker hub repository.

```

root@DESKTOP-800G2HF:kubernetes-hpa-example# docker image build -t asfiyask/cloudethix_hpa_asfiya:vh . --load
[*] Building 8.9s (11/11) FINISHED                                docker-container:strange_perlman
=> [internal] load build definition from Dockerfile                0.1s
=> => transferring dockerfile: 164B                                0.0s
=> [internal] load metadata for docker.io/library/node:8.12.0-alpine 0.8s
=> [internal] load .dockerignore                                   0.0s
=> => transferring context: 2B                                       0.0s
=> [1/5] FROM docker.io/library/node:8.12.0-alpine@sha256:d75742c5fd41261113ed4706f961a21238db84648c825a5126ada373c361f46e 0.1s
=> => resolve docker.io/library/node:8.12.0-alpine@sha256:d75742c5fd41261113ed4706f961a21238db84648c825a5126ada373c361f46e 0.1s
=> [internal] load build context                                   0.8s
=> => transferring context: 2.12kB                                    0.8s
=> CACHED [2/5] RUN mkdir -p /usr/src/app                         0.0s
=> CACHED [3/5] WORKDIR /usr/src/app                              0.0s
=> CACHED [4/5] COPY . /usr/src/app                               0.0s
=> CACHED [5/5] RUN npm install                                    0.0s
=> exporting to docker image format                               6.8s
=> => exporting layers                                              0.5s
=> => exporting manifest sha256:4f5d796996bfdd6ad3c53298d89220cba2e21b6f8c408f8bd0103813dcca9a2 0.0s
=> => exporting config sha256:37bb4aed96bc6e4a3bea251df901746e5f079e79132268eded1485e52ec41c33 0.0s
=> => sending tarball                                              6.3s
=> importing to docker                                           5.8s

```

```

root@DESKTOP-800G2HF:kubernetes-hpa-example# docker image ls

```

REPOSITORY	TAG	IMAGE
ID	CREATED	SIZE
asfiyask/cloudethix_hpa_asfiya	vh	
37bb4aed96bc	50 seconds ago	65.7MB

```

root@DESKTOP-800G2HF:kubernetes-hpa-example# docker image push
asfiyask/cloudethix_hpa_asfiya:vh

```

The push refers to repository
[docker.io/asfiyask/cloudethix_hpa_asfiya]

b8e9c0a07126: Pushed

77abe5d8f1bf: Pushed

5f70bf18a086: Mounted from asfiyask/kubernetes-haproxy-nodejs-service

f4d297a2cf06: Pushed

8b59e4cead98: Pushed

7aa09d2ca0a3: Pushed

df64d3292fd6: Pushed

vh: digest:

sha256:3d06d57b00748f65426eeb58f6b08ca5e724559bc3fc3bca22c4ed9e8a75fe4
0 size: 1780

-
- Once the image is pushed, go to k8s directory and update deployment.yaml file with image name from your repo. And then create it.

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: node-example
spec:
  replicas: 1
  selector:

```

```

matchLabels:
  app: node-example
template:
  metadata:
    labels:
      app: node-example
  spec:
    containers:
      - name: node-example
        image: asfiyask/cloudethix_hpa_asfiya:vh
        imagePullPolicy: Always
        ports:
          - containerPort: 3000
        resources:
          limits:
            cpu: "0.5"
          requests:
            cpu: "0.25"

```

- Open service.yml and change the type to nodePort and apply the same.

```

apiVersion: v1
kind: Service
metadata:
  name: node-example
  labels:
    app: node-example
spec:
  selector:
    app: node-example
  ports:
    - port: 80
      protocol: TCP
      targetPort: 3000
      nodePort: 30111
  type: NodePort

```

- Open the HPA.yaml file, notice it and then apply the same.

```
root@DESKTOP-800G2HF:k8s# k apply -f .
```

```
deployment.apps/node-example created
```

```
horizontalpodautoscaler.autoscaling/node-example created
```

```
service/node-example created
```

```
root@DESKTOP-800G2HF:k8s# kgp
```

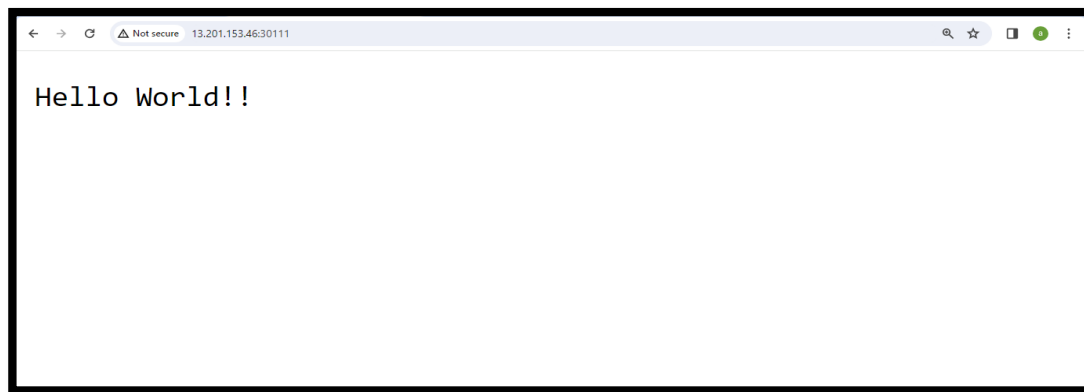
NAME	READY	STATUS	RESTARTS	AGE
node-example-688488b678-vk54q	1/1	Running	0	26s

```
root@DESKTOP-800G2HF:k8s# kgs
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
node-example	NodePort	10.96.90.172	<none>	3000:30111/TCP

28s

-
- Open the browser, and access the webpage



-
- Now it's time to test the HPA working with the below command.

```
# kubectl run -i --tty load-generator --rm --image=busybox  
--restart=Never -- /bin/sh -c "while sleep 0.01; do wget -q -  
Ohttp://NODE_PORT_SERVICE_NAME; done"
```

```
root@DESKTOP-800G2HF:k8s# kubectl run -i --tty load-generator --rm --  
image=busybox --restart=Never -- /bin/sh -c "while sleep 0.01; do wget  
-q -O- http://node-example:80; done"
```

If you don't see a command prompt, try pressing enter.

Hello World!!

Hello World!!

Hello World!!

Hello World!!

Hello World!!

Hello World!!

-
- Check the HPA from kubectl command and also check if the deployment is scaling up.

```
root@DESKTOP-800G2HF:k8s# kubectl get hpa
```

NAME	REFERENCE	TARGETS	MINPODS
MAXPODS	REPLICAS	AGE	
node-example	Deployment/node-example	<unknown>/1%	1
0	27m		4

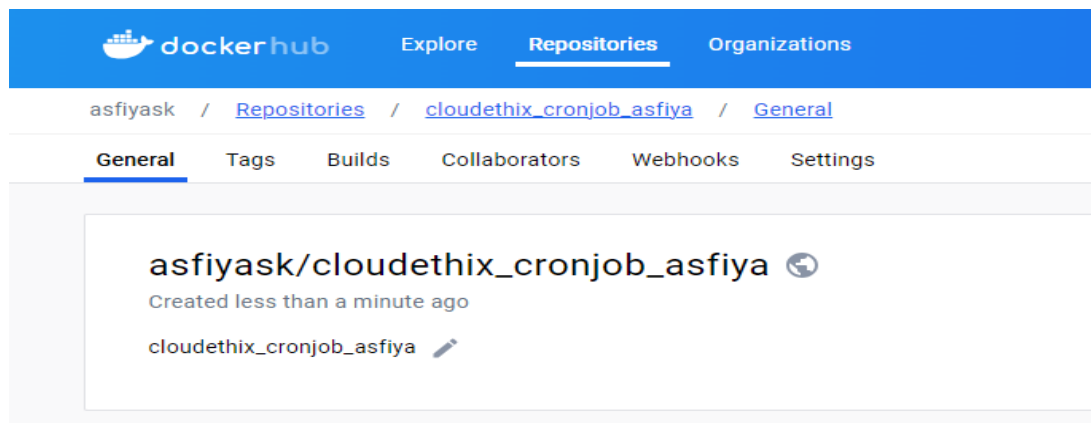
```
root@DESKTOP-800G2HF:k8s# kubectl get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
node-example	1/1	1	1	27m

=====

Que 14 →

- Create 1 Public Docker Hub registry named cloudethix_cronjob_yourname.



```
root@DESKTOP-800G2HF:my_scripts# mkdir Cronjob
```

```
root@DESKTOP-800G2HF:my_scripts# cd Cronjob/
```

```
root@DESKTOP-800G2HF:Cronjob# ll
```

```
total 0
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 23 11:22 ./
```

```
drwxrwxrwx 1 asfiya asfiya 512 Feb 23 11:22 ../
```

```
root@DESKTOP-800G2HF:Cronjob# touch helloworld.py
```

```
# Hellowold.py
#!/usr/local/bin/python3
import datetime
x = datetime.datetime.now()
print("Welcome to the Cloudethix World")
print("Today is")
print(x)
```

```
root@DESKTOP-800G2HF:Cronjob# touch Dockerfile
```

```
# Dockerfile
FROM python:3.7-alpine
#add user group and ass user to that group
RUN addgroup -S appgroup && adduser -S appuser -G appgroup
#creates work dir
WORKDIR /app
#copy python script to the container folder app
COPY helloworld.py /app/helloworld.py
RUN chmod +x /app/helloworld.py
#user is appuser
USER appuser
ENTRYPOINT ["python", "/app/helloworld.py"]
```

```
root@DESKTOP-800G2HF:Cronjob# touch pythoncronjob.yml
```

```
root@DESKTOP-800G2HF:Cronjob# docker image build -t asfiyask/cloudethix_cronjob_asfiya:v1 . --load
[+] Building 7.7s (11/11) FINISHED                                docker-container: strange_perlman
=> [internal] load build definition from Dockerfile                0.1s
=> => transferring dockerfile: 396B                                0.0s
=> [internal] load metadata for docker.io/library/python:3.7-alpine 0.8s
=> [internal] load .dockerignore                                  0.1s
=> => transferring context: 2B                                       0.0s
=> [1/5] FROM docker.io/library/python:3.7-alpine@sha256:f3d31c8677d03f0b3c724446077f229a6ce9d3ac430f5c08cd7dffa00292048c3 0.1s
=> => resolve docker.io/library/python:3.7-alpine@sha256:f3d31c8677d03f0b3c724446077f229a6ce9d3ac430f5c08cd7dffa00292048c3 0.1s
=> [internal] load build context                                  0.1s
=> => transferring context: 32B                                       0.0s
=> CACHED [2/5] RUN addgroup -S appgroup && adduser -S appuser -G appgroup 0.0s
=> CACHED [3/5] WORKDIR /app                                       0.0s
=> CACHED [4/5] COPY helloworld.py /app/helloworld.py             0.0s
=> CACHED [5/5] RUN chmod +x /app/helloworld.py                   0.0s
=> exporting to docker image format                               6.3s
=> => exporting layers                                              1.0s
=> => exporting manifest sha256:505a04fala343d5e11d18679eedd04d459ce8b289e24971300b99e7dec62c58f 0.1s
=> => exporting config sha256:3fe0da619649e9e8081214d735ae060b69da97da301cbf5c1ed4787742f4520a 0.0s
=> => sending tarball                                              5.2s
=> importing to docker                                           4.4s
```

```
root@DESKTOP-800G2HF:Cronjob# docker image push
asfiyask/cloudethix_cronjob_asfiya:v1
```

The push refers to repository

[docker.io/asfiyask/cloudethix_cronjob_asfiya]

5f70bf18a086: Mounted from asfiyask/cloudethix_hpa_asfiya

68ff62071c60: Pushed

4daa97c307e6: Pushed

cc3791a4c448: Pushed

ae2ed3079163: Pushed

aa3a591fc84e: Pushed

7f29b11ef9dd: Pushed

a1c2f058ec5f: Pushed

cc2447e1835a: Pushed

v1: digest:

sha256:c50a6a105b39ef3d5df62558f3f4e44b034416ce26f478cc93c8500f8902fced size: 2195

-
- Now update the pythoncronjob.yml file to change the image name that you have just pushed to docker hub registry.

```
# Python Cronjob Yaml
apiVersion: batch/v1
kind: CronJob
metadata:
  name: python-helloworld
spec:
  schedule: "*/1 * * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
            - name: python-helloworld
              image: asfiyask/cloudethix_cronjob_asfiya:v1
              command: ["/app/helloworld.py"]
          restartPolicy: OnFailure
```

-
- Now create a cron job using pythoncronjob.yml file. Check with kubectl command if the cron job is created.

```
root@DESKTOP-800G2HF:Cronjob# k apply -f pythoncronjob.yml
```

```
cronjob.batch/python-helloworld created
```

```
root@DESKTOP-800G2HF:Cronjob# kgp
```

```
No resources found in default namespace.
```



```
root@DESKTOP-800G2HF:Cronjob# kubectl get cronjob
```

NAME AGE	SCHEDULE	SUSPEND	ACTIVE	LAST SCHEDULE
python-helloworld 72s	*/1 * * * *	False	1	46s

-
- Then check the pod logs which are created by the job and capture the output.

```
root@DESKTOP-800G2HF:14_Cronjob# kgp
```

NAME	READY	STATUS	RESTARTS	AGE
python-helloworld-28478206-5bzfz	0/1	Completed	0	22s

```
root@DESKTOP-800G2HF:14_Cronjob# k logs python-helloworld-28478206-5bzfz
```

```
Welcome to the Cloudethix World
```

```
Today is
```

```
2024-02-23 12:46:07.011640
```

```
root@DESKTOP-800G2HF:14_Cronjob# k get cronjob
```

NAME AGE	SCHEDULE	SUSPEND	ACTIVE	LAST SCHEDULE
python-helloworld 98s	*/1 * * * *	False	0	59s

```
apiVersion: batch/v1
kind: CronJob
metadata:
  name: python-helloworld
spec:
  schedule: "*/1 * * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
            - name: python-helloworld
              image: asfiyask/cronjob_asfiya:v1
              command: ["/app/helloworld.py"]
          restartPolicy: OnFailure
```

