

CLOUDETHIX

Que 1 →

SOLUTION:-

- Create 2 Public Docker Hub registries named `cloudeithix_master_nginx_yourname` & `cloudeithix_release_nginx_yourname`.

The image contains three screenshots of Docker Hub interfaces:

- Top Screenshot:** Shows the 'General' tab for the repository `sanu28221/cloudeithix_master_nginx_saniya`. It includes fields for adding a short description, Docker commands (with a placeholder for `docker push sanu28221/cloudeithix_master_nginx_saniya:tagname`), and an 'Automated Builds' section.
- Middle Screenshot:** Shows the 'General' tab for the repository `sanu28221/cloudeithix_release_nginx_saniya`, similar to the top one but with a different repository name.
- Bottom Screenshot:** Shows the user profile page for `sanu28221`, displaying two repositories: `cloudeithix_release_nginx_saniya` and `cloudeithix_master_nginx_saniya`. Both repositories are listed as containing 'No content' and were created '2 minutes ago'. The 'Create repository' button is visible at the top right of the profile page.

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

```
root@DESKTOP-VIDGD8F:Assignment_1# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 21 12:27 .
drwxrwxrwx 1 saniya saniya 512 Feb 21 12:27 ..
drwxrwxrwx 1 saniya saniya 512 Feb 21 12:27 .git
root@DESKTOP-VIDGD8F:Assignment_1# 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-VIDGD8F:Assignment_1# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 21 12:27 .
drwxrwxrwx 1 saniya saniya 512 Feb 21 12:27 ..
drwxrwxrwx 1 saniya saniya 512 Feb 21 12:27 .git
drwxrwxrwx 1 saniya saniya 512 Feb 21 12:27 docker-sample-nginx
root@DESKTOP-VIDGD8F:Assignment_1# █
```

- 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
```

```
root@DESKTOP-VIDGD8F:docker-sample-nginx# git branch
* master
root@DESKTOP-VIDGD8F:docker-sample-nginx# git branch release
root@DESKTOP-VIDGD8F:docker-sample-nginx# git branch main
root@DESKTOP-VIDGD8F:docker-sample-nginx# git branch hotfix
root@DESKTOP-VIDGD8F:docker-sample-nginx# git branch
  hotfix
  main
* master
  release
```

- 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 cloudeithix_master_nginx_yourname.

```
root@DESKTOP-VIDGD8F:docker-sample-nginx# cat index.html
<html>
<body>
    <h1>Host: <!--#echo var="HOSTNAME" --></h1>
    Version: 1.1
</body>
</html>
root@DESKTOP-VIDGD8F:docker-sample-nginx# vim index.html
```

```

root@DESKTOP-VIDGD8F:docker-sample-nginx# cat index.html
"Cloudethix Master Branch Nginx"
root@DESKTOP-VIDGD8F:docker-sample-nginx# 

root@DESKTOP-VIDGD8F:docker-sample-nginx# docker build -t sanu28221/cloudethix_master_nginx_saniya:master_image . --no-cache
[+] Building 40.7s (9/9) FINISHED                                            docker:default
--> [internal] load build definition from Dockerfile
--> => transferring dockerfile: 122B
--> [internal] load dockerignore
--> => transferring context: 28
--> [internal] load metadata for docker.io/library/nginx:alpine
--> [auth] library/nginx pull token for registry-1.docker.io
--> [1/3] FROZEN docker.io/library/nginx:alpine@sha256:6a7fb528e45c4adea04ec207a251fd4a2df03ddc930f782af51e315ebc76e9a9
--> => resolve_docker_lo[library/nginx:alpine@sha256:6a7fb528e45c4adea04ec207a251fd4a2df03ddc930f782af51e315ebc76e9a9]
--> => sha256:6a7fb528e45c4adea04ec207a251fd4a2df03ddc930f782af51e315ebc76e9a9 8.71kB / 8.71kB
--> => sha256:6913ed9ec8d009744801ac178087027fe2a085935b2cc7a224bf85347b670d7 11.70kB / 11.70kB
--> => sha256:6913ed9ec8d009744801ac178087027fe2a085935b2cc7a224bf85347b670d7 11.70kB / 11.70kB
--> => sha256:c1ea3344e7111de0e02f17decbea7235ed10wes99bf7fb472114dc1399ce 629B / 629B
--> => sha256:c1ea3344e7111de0e02f17decbea7235ed10wes99bf7fb472114dc1399ce 629B / 629B
--> => sha256:018b9865f459b5c2227ae979a49a2284946d997fa4d4ed7a139ebcb6aef2e70B 2.30kB / 2.30kB
--> => sha256:018b9865f459b5c2227ae979a49a2284946d997fa4d4ed7a139ebcb6aef2e70B 2.30kB / 2.30kB
--> => sha256:018b9865f459b5c2227ae979a49a2284946d997fa4d4ed7a139ebcb6aef2e70B 1.98kB / 1.98kB
--> => extracting sha256:018b9865f459b5c2227ae979a49a2284946d997fa4d4ed7a139ebcb6aef2e70B 1.98kB / 1.98kB
--> => sha256:d6x45d092aaNc8032894fc3da9929f31c505232fcf1925db314815a196c00ff 1.21kB / 1.21kB
--> => sha256:a19ic9a82bb8a3fa561030af162d98a130ca1abc0501b7e785944fbdd0f2c96 393B / 393B
--> => sha256:c7059f1102784cd05d96ff7fa459bc9fa50ff8a724ec8d748507fa3455999b 956B / 956B
--> => extracting sha256:018b9865f459b5c2227ae979a49a2284946d997fa4d4ed7a139ebcb6aef2e70B 12.65MB
--> => sha256:a1cc681b03a03fff27eccf0ffcc152e0b1bcc2e00c0c9e371b58f45680d05c1954 1.40kB / 1.40kB
--> => extracting sha256:c17859f1102784cd05d96ff7fa459bc9fa50ff8a724ec8d748507fa3455999b
--> => extracting sha256:c17859f1102784cd05d96ff7fa459bc9fa50ff8a724ec8d748507fa3455999b
--> => extracting sha256:d0a456972aa0c0803189ffec3da0939f31c5885232fcf1925db314815a196c00ff
--> => extracting sha256:e1cc681b03a03fff27eccf0ffcc152e0b1bcc2e00c0c9e371b58f45680d05c1954
--> => extracting sha256:a1cc681b03a03fff27eccf0ffcc152e0b1bcc2e00c0c9e371b58f45680d05c1954
--> [internal] load build context
--> => transferring context: 491B
--> [2/3] COPY default.conf /etc/nginx/conf.d/
--> [3/3] COPY index.html /usr/share/nginx/html/
--> => exporting to image
--> => writing image sha256:8607d58001ae4418ccaa50d8f9401ca2c0d8ef6b4d07cf5c952656630507c89
--> => naming to docker.io/sanu28221/cloudethix_master_nginx_saniya:master_image
root@DESKTOP-VIDGD8F:docker-sample-nginx# 

```

Activate Windows

```

root@DESKTOP-VIDGD8F:docker-sample-nginx# docker image push sanu28221/cloudethix_master_nginx_saniya:master_image
The push refers to repository [docker.io/sanu28221/cloudethix_master_nginx_saniya]
4a994503a66e: Pushed
600996b88f91: 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 library/nginx
718db50a47c0: Mounted from library/nginx
aedc3bda2944: Mounted from library/nginx
master_image: digest: sha256:3bd7f723d8d5b5ae4f0e959bb8a9b0950244b9743389cda79fb7ad804d94bc6a size: 2403
root@DESKTOP-VIDGD8F:docker-sample-nginx# 

```

sanu28221/cloudethix_master_nginx_saniya

Updated 7 minutes ago

This repository does not have a description

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
 master_image		Image	--	7 minutes ago

[See all](#)

- 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.

```
root@DESKTOP-VIDGD8F:docker-sample-nginx# git checkout release
M      index.html
Switched to branch 'release'
root@DESKTOP-VIDGD8F:docker-sample-nginx# ll
total 4
drwxrwxrwx 1 saniya saniya 512 Feb 21 12:36 .
drwxrwxrwx 1 saniya saniya 512 Feb 21 12:27 ..
drwxrwxrwx 1 saniya saniya 512 Feb 21 12:50 .git
-rwxrwxrwx 1 saniya saniya 95 Feb 21 12:27 Dockerfile
-rwxrwxrwx 1 saniya saniya 1084 Feb 21 12:27 LICENSE
-rwxrwxrwx 1 saniya saniya 73 Feb 21 12:27 README.md
-rwxrwxrwx 1 saniya saniya 286 Feb 21 12:27 default.conf
-rwxrwxrwx 1 saniya saniya 125 Feb 21 12:36 index.html
root@DESKTOP-VIDGD8F:docker-sample-nginx# cat index.html
<html>
<body>
    <h1>Host: <!--#echo var="Cloudethix Master Branch Nginx" --></h1>
    Version: 1.1
</body>
</html>
root@DESKTOP-VIDGD8F:docker-sample-nginx# vim index.html
root@DESKTOP-VIDGD8F:docker-sample-nginx# cat index.html
"Cloudethix Release Branch Nginx"
root@DESKTOP-VIDGD8F:docker-sample-nginx#
```

```
root@DESKTOP-VIDGD8F:docker-sample-nginx# docker build -t sanu28221/cloudethix_release_nginx_saniya:release_image . --no-cache
[+] Building 35.3s (9/9) FINISHED
=> [internal] load dockerignore
=> [internal] load context: 2B
=> [internal] load build definition from Dockerfile
=> [internal] load build context
=> [internal] load metadata for docker.io/library/nginx:alpine
=> [auth] library/nginx:pull token for registry-1.docker.io
=> [CACHED] [1/2] FROM docker.io/library/nginx:alpine@sha256:6a2f8b28e45c4adea04ec287a251fd4a2d03ddc930f782af51e315ebc76e9a9
=> [internal] load build context
=> [internal] load context: 1998
=> [2/2] COPY default.conf /etc/nginx/conf.d/
=> [3/3] COPY index.html /usr/share/nginx/html/
=> exporting to image
=> exporting layers
=> writing image sha256:78504fdc05669589a28ae83b46109772d0d254fd507870089cd01741178c01cff
=> naming to docker.io/sanu28221/cloudethix_release_nginx_saniya:release_image
root@DESKTOP-VIDGD8F:docker-sample-nginx#
```

```
root@DESKTOP-VIDGD8F:docker-sample-nginx# docker image push sanu28221/cloudethix_release_nginx_saniya:release_image
The push refers to repository [docker.io/sanu28221/cloudethix_release_nginx_saniya]
4ac8f920dc28: Pushed
6d2db469fa96: Pushed
667a247707f0: Mounted from sanu28221/cloudethix_master_nginx_saniya
d8527026595f: Mounted from sanu28221/cloudethix_master_nginx_saniya
2593b08e5428: Mounted from sanu28221/cloudethix_master_nginx_saniya
9909978d630d: Mounted from sanu28221/cloudethix_master_nginx_saniya
c5140fc719dd: Mounted from sanu28221/cloudethix_master_nginx_saniya
3137f8fc641: Mounted from sanu28221/cloudethix_master_nginx_saniya
718db50a47c0: Mounted from sanu28221/cloudethix_master_nginx_saniya
aedc3bda2944: Mounted from sanu28221/cloudethix_master_nginx_saniya
release_image: digest: sha256:e0dc19df1f5e28d00ad53b5bd382d05664c7014ff8fd778b7c9ef0b077500010 size: 2403
root@DESKTOP-VIDGD8F:docker-sample-nginx#
```

sanu28221/cloudethix_release_nginx_saniya

Updated 1 minute ago

This repository does not have a description

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
 release_image		Image	---	a few seconds ago

[See all](#)

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

```
root@DESKTOP-VIDGD8F:docker-sample-nginx# git checkout main
M      index.html
Switched to branch 'main'
root@DESKTOP-VIDGD8F:docker-sample-nginx#
```

- 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.

```
root@DESKTOP-VIDGD8F:docker-sample-nginx# git branch
  hotfix
* main
  master
  release
root@DESKTOP-VIDGD8F:docker-sample-nginx# ll
total 4
drwxrwxrwx 1 saniya saniya 512 Feb 21 12:51 .
drwxrwxrwx 1 saniya saniya 512 Feb 21 12:27 ..
drwxrwxrwx 1 saniya saniya 512 Feb 21 12:59 .git
-rwxrwxrwx 1 saniya saniya 95 Feb 21 12:27 Dockerfile
-rwxrwxrwx 1 saniya saniya 1084 Feb 21 12:27 LICENSE
-rwxrwxrwx 1 saniya saniya 73 Feb 21 12:27 README.md
-rwxrwxrwx 1 saniya saniya 286 Feb 21 12:27 default.conf
-rwxrwxrwx 1 saniya saniya 126 Feb 21 12:51 index.html
root@DESKTOP-VIDGD8F:docker-sample-nginx# mkdir kube
root@DESKTOP-VIDGD8F:docker-sample-nginx# mkdir clusterIP
```

```
root@DESKTOP-VIDGD8F:docker-sample-nginx# cd kube/
root@DESKTOP-VIDGD8F:kube# touch master_pod.yaml
root@DESKTOP-VIDGD8F:kube# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 21 13:05 .
drwxrwxrwx 1 saniya saniya 512 Feb 21 13:05 ..
-rwxrwxrwx 1 saniya saniya 267 Feb 21 13:09 master_pod.yaml
```

```
root@DESKTOP-VIDGD8F:kube# cat master_pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: master-nginx
  labels:
    name: master-nginx
spec:
  containers:
  - name: master-nginx-cont
    image: sanu28221/cloudeithix_master_nginx_saniya:master_image
    resources:
    ports:
      - containerPort: 80root@DESKTOP-VIDGD8F:kube# █
```

- 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 cloudeithix_release_nginx_yourname.

```
root@DESKTOP-VIDGD8F:kube# touch release_pod.yaml
root@DESKTOP-VIDGD8F:kube# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 21 13:12 .
drwxrwxrwx 1 saniya saniya 512 Feb 21 13:05 ..
-rwxrwxrwx 1 saniya saniya 267 Feb 21 13:09 master_pod.yaml
-rwxrwxrwx 1 saniya saniya 0 Feb 21 13:12 release_pod.yaml
root@DESKTOP-VIDGD8F:kube# cat release_pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: release-nginx
  labels:
    name: release-nginx
spec:
  containers:
  - name: release-nginx-cont
    image: sanu28221/cloudeithix_release_nginx_saniya:release_image
    resources:
    ports:
      - containerPort: 80root@DESKTOP-VIDGD8F:kube# █
root@DESKTOP-VIDGD8F:kube# █
```

- Create a file called cluster_ip-service.yaml with service name cloudeithix_clusterip and with Type clusterIP.
- Then, select the pod with label release_nginx in service.

```
root@DESKTOP-VIDGD8F:kube# touch cluster_ip-service.yaml
root@DESKTOP-VIDGD8F:kube# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 21 13:17 .
drwxrwxrwx 1 saniya saniya 512 Feb 21 13:05 ..
-rwxrwxrwx 1 saniya saniya 0 Feb 21 13:17 cluster_ip-service.yaml
-rwxrwxrwx 1 saniya saniya 265 Feb 21 13:14 master_pod.yaml
-rwxrwxrwx 1 saniya saniya 270 Feb 21 13:14 release_pod.yaml
```

```
root@DESKTOP-VIDGD8F:kube# cat cluster_ip-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: cloudeithix-clusterip
spec:
  type: ClusterIP
  selector:
    app: release-nginx
  ports:
  - port: 80
    targetPort: 80
root@DESKTOP-VIDGD8F:kube# █
```

- Create all these three resources in your k8s cluster.

```
root@DESKTOP-VIDGD8F:kube# kcf .
service/cloudeithix-clusterip created
pod/master-nginx created
pod/release-nginx created
```

```
root@DESKTOP-VIDGD8F:kube# kgp
NAME        READY   STATUS    RESTARTS   AGE
master-nginx 1/1     Running   0          3m38s
release-nginx 1/1     Running   0          3m38s
root@DESKTOP-VIDGD8F:kube# kgs
NAME           TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
cloudeithix-clusterip  ClusterIP  10.99.178.175  <none>        80/TCP      3m40s
kubernetes     ClusterIP  10.96.0.1     <none>        443/TCP     143m
root@DESKTOP-VIDGD8F:kube# █
```

- Now, access master_nginx pod shell & curl the master_nginx pod & check the result.

```
root@DESKTOP-VIDGD8F:kube# kubectl exec -it master-nginx -- /bin/sh
/ # curl localhost
Cloudethix Master Branch Nginx
/ # █
```

- Also try to curl release_nginx pod with DNS name & check the result.

```
root@DESKTOP-VIDGD8F:kube# kubectl exec -it release-nginx -- /bin/sh
/ # curl localhost
Cloudethix Release Branch Nginx
/ #
```

- Then curl the clusterip service with its name and check the result.

```
root@DESKTOP-VIDGD8F:kube# kubectl exec -it master-nginx -- /bin/sh
/ # curl cloudethix-clusterip
"Cloudethix Release Branch Nginx"
/ #
```

- Finally, create a GITHUB remote repository named cloudehix-k8s-yourname and push all the branches to the remote repository.

The screenshot shows a GitHub repository page. At the top, it displays the repository name 'Saniya2822 / cloudehix-k8s-saniya'. Below the header, there are navigation links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Code' tab is selected. The main content area shows a single branch named 'main' with 1 branch and 0 tags. A file named 'README.md' is listed, showing its content: 'Initial commit'. The commit was made by 'Saniya2822' at '13c2a43 · now' with 1 commit. Below the file listing, the 'README' file is shown with the text 'cloudehix-k8s-saniya'.

Your branches					
Branch	Updated	Check status	Behind	Ahead	Pull request
release	5 minutes ago	Green	1 8	8	...
main	5 minutes ago	Green	1 10	10	...
hotfix	6 minutes ago	Green	1 7	7	...
master	6 minutes ago	Green	1 9	9	...

Active branches					
Branch	Updated	Check status	Behind	Ahead	Pull request
release	5 minutes ago	Green	1 8	8	...
main	5 minutes ago	Green	1 10	10	...
hotfix	6 minutes ago	Green	1 7	7	...
master	6 minutes ago	Green	1 9	9	...

Activate Windows

- Take all screenshots and create a well formatted document.

Que 2 →

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

```
root@DESKTOP-VIDGD8F:kube# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 21 16:32 .
drwxrwxrwx 1 saniya saniya 512 Feb 21 16:13 ..
drwxrwxrwx 1 saniya saniya 512 Feb 21 16:32 clusterip
root@DESKTOP-VIDGD8F:kube# mkdir NodePort
root@DESKTOP-VIDGD8F:kube# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 21 16:35 .
drwxrwxrwx 1 saniya saniya 512 Feb 21 16:13 ..
drwxrwxrwx 1 saniya saniya 512 Feb 21 16:35 NodePort
drwxrwxrwx 1 saniya saniya 512 Feb 21 16:32 clusterip
root@DESKTOP-VIDGD8F:kube# █
```

- 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/connecting-frontend-backend/>.

```
backend-deployment.yaml
backend-service.yaml
frontend-deployment.yaml
frontend-NodePort-service.yaml
```

```
root@DESKTOP-VIDGD8F:NodePort# ll
total 1
drwxrwxrwx 1 saniya saniya 512 Feb 21 16:39 .
drwxrwxrwx 1 saniya saniya 512 Feb 21 16:35 ..
-rwxrwxrwx 1 saniya saniya 476 Feb 21 16:39 backend-deployment.yaml
-rwxrwxrwx 1 saniya saniya 175 Feb 21 16:40 backend-service.yaml
-rwxrwxrwx 1 saniya saniya 218 Feb 21 16:41 frontend-NodePort-service.yaml
-rwxrwxrwx 1 saniya saniya 527 Feb 21 16:40 frontend-deployment.yaml
root@DESKTOP-VIDGD8F:NodePort# █
```

```
root@DESKTOP-VIDGD8F:NodePort# cat backend-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: backend
spec:
  selector:
    matchLabels:
      app: hello
      tier: backend
      track: stable
  replicas: 3
  template:
    metadata:
      labels:
        app: hello
        tier: backend
        track: stable
    spec:
      containers:
        - name: hello
          image: "gcr.io/google-samples/hello-go-gke:1.0"
          ports:
            - name: http
              containerPort: 80root@DESKTOP-VIDGD8F:NodePort# █
```

```
root@DESKTOP-VIDGD8F:NodePort# cat backend-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: hello
spec:
  selector:
    app: hello
    tier: backend
  ports:
    - protocol: TCP
      port: 80
      targetPort: httproot@DESKTOP-VIDGD8F:NodePort# █
```

```
root@DESKTOP-VIDGD8F:NodePort# cat frontend-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend
spec:
  selector:
    matchLabels:
      app: hello
      tier: frontend
      track: stable
  replicas: 1
  template:
    metadata:
      labels:
        app: hello
        tier: frontend
        track: stable
    spec:
      containers:
        - name: nginx
          image: "gcr.io/google-samples/hello-frontend:1.0"
          lifecycle:
            preStop:
              exec:
                command: ["/usr/sbin/nginx", "-s", "quit"]
root@DESKTOP-VIDGD8F:NodePort#
```

```
root@DESKTOP-VIDGD8F:NodePort# cat frontend-NodePort-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: frontend
spec:
  selector:
    app: hello
    tier: frontend
  ports:
    - protocol: "TCP"
      port: 80
      targetPort: 80
      nodePort: 30008
  type: NodePort
root@DESKTOP-VIDGD8F:NodePort#
```

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

```
root@DESKTOP-VIDGD8F:NodePort# kcf .
deployment.apps/backend created
service/hello created
service/frontend created
deployment.apps/frontend created
```

```
root@DESKTOP-VIDGD8F:NodePort# kgp
NAME                  READY   STATUS    RESTARTS   AGE
backend-7f5b7998b9-4dc9v   1/1     Running   0          20s
backend-7f5b7998b9-cv89b   1/1     Running   0          20s
backend-7f5b7998b9-l4xrk   1/1     Running   0          20s
frontend-85c84f8b8b-d47rk  1/1     Running   0          20s
master-nginx              1/1     Running   0          63m
release-nginx              1/1     Running   0          52m
root@DESKTOP-VIDGD8F:NodePort#
```

```
root@DESKTOP-VIDGD8F:NodePort# kgs
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
cloudethix-clusterip ClusterIP  10.103.195.1   <none>           80/TCP        52m
frontend       NodePort    10.102.226.115  <none>           80:30008/TCP  8s
hello          ClusterIP  10.102.180.156  <none>           80/TCP        8s
kubernetes     ClusterIP  10.96.0.1       <none>           443/TCP      5h41m
```

- 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.
- Take all screenshots and create a well formatted document.

Que 3 →

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

SOLUTION:-

```
root@DESKTOP-VIDGD8F:KUBE# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 21 17:08 ..
drwxrwxrwx 1 saniya saniya 512 Feb 21 17:08 ..
root@DESKTOP-VIDGD8F:KUBE# touch pod_01.yaml
root@DESKTOP-VIDGD8F:KUBE# touch pod_02.yaml
```

```
root@DESKTOP-VIDGD8F:KUBE# cat pod_01.yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  nodeName: worker-1
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
      - containerPort: 80root@DESKTOP-VIDGD8F:KUBE#
```

```
root@DESKTOP-VIDGD8F:KUBE# cat pod_02.yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  nodeName: worker-0
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
      - containerPort: 80root@DESKTOP-VIDGD8F:KUBE#
root@DESKTOP-VIDGD8F:KUBE# █
```

Que 4 →

- Label both worker nodes such as worker-0 node as cloudeithix-k8s-00 & worker-1 node as cloudeithix-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-VIDGD8F:Ques4# kubectl label node worker-0 Name=cloudeithix-k8s-00  
node/worker-0 labeled
```

```
root@DESKTOP-VIDGD8F:Ques4# kubectl label node worker-1 Name=cloudeithix-k8s-01  
node/worker-1 labeled  
root@DESKTOP-VIDGD8F:Ques4#
```

```
root@DESKTOP-VIDGD8F:Ques4# ll  
total 0  
drwxrwxrwx 1 saniya saniya 512 Feb 21 17:45 .  
drwxrwxrwx 1 saniya saniya 512 Feb 21 17:45 ..  
root@DESKTOP-VIDGD8F:Ques4# touch pod00.yaml  
root@DESKTOP-VIDGD8F:Ques4# touch pod01.yaml  
root@DESKTOP-VIDGD8F:Ques4# ll  
total 0  
drwxrwxrwx 1 saniya saniya 512 Feb 21 18:06 .  
drwxrwxrwx 1 saniya saniya 512 Feb 21 17:45 ..  
-rwxrwxrwx 1 saniya saniya 158 Feb 21 18:10 pod00.yaml  
-rwxrwxrwx 1 saniya saniya 158 Feb 21 18:10 pod01.yaml
```

```
root@DESKTOP-VIDGD8F:Ques4# cat pod00.yaml  
apiVersion: v1  
kind: Pod  
metadata:  
  name: nginx-bigdata  
spec:  
  containers:  
  - name: data-nginx  
    image: nginx  
  nodeSelector:  
    Name: cloudeithix-k8s-00  
root@DESKTOP-VIDGD8F:Ques4#
```

```

root@DESKTOP-VIDGD8F:Ques4# cat pod01.yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-bigdata
spec:
  containers:
  - name: data-nginx
    image: nginx
  nodeSelector:
    Name: cloudehix-k8s-01
root@DESKTOP-VIDGD8F:Ques4# █

```

Que 5 →

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

SOLUTION:-

```

root@DESKTOP-VIDGD8F:Ques5# git clone https://github.com/collabnix/kubelabs
Cloning into 'kubelabs'...
remote: Enumerating objects: 12313, done.
remote: Counting objects: 100% (1164/1164), done.
remote: Compressing objects: 100% (486/486), done.
remote: Total 12313 (delta 699), reused 1050 (delta 643), pack-reused 11149
Receiving objects: 100% (12313/12313), 61.41 MiB | 9.42 MiB/s, done.
Resolving deltas: 100% (3275/3275), done.
Updating files: 100% (7329/7329), done.
root@DESKTOP-VIDGD8F:Ques5#
root@DESKTOP-VIDGD8F:Ques5# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 21 18:12 .
drwxrwxrwx 1 saniya saniya 512 Feb 21 18:12 ..
drwxrwxrwx 1 saniya saniya 512 Feb 21 18:12 .git
drwxrwxrwx 1 saniya saniya 512 Feb 21 18:15 kubelabs
root@DESKTOP-VIDGD8F:Ques5# █

```

```

root@DESKTOP-VIDGD8F:kubelabs# cd DaemonSet101
root@DESKTOP-VIDGD8F:DaemonSet101# ll
total 8
drwxrwxrwx 1 saniya saniya 512 Feb 21 18:15 .
drwxrwxrwx 1 saniya saniya 512 Feb 21 18:15 ..
-rwxrwxrwx 1 saniya saniya 7040 Feb 21 18:15 README.md
-rwxrwxrwx 1 saniya saniya 394 Feb 21 18:15 daemonset.yml
root@DESKTOP-VIDGD8F:DaemonSet101# kubectl apply -f daemonset.yml
daemonset.apps/prometheus-daemonset created
root@DESKTOP-VIDGD8F:DaemonSet101# kubectl get daemonsets
NAME           DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
prometheus-daemonset   2         2         2         2           2           <none>      49s
root@DESKTOP-VIDGD8F:DaemonSet101# █

```

Que 6 →

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

SOLUTION:-

```
root@DESKTOP-VIDGD8F:Ques6# ssh -i ~/.ssh/id_rsa ubuntu@3.111.168.44
The authenticity of host '3.111.168.44 (3.111.168.44)' can't be established.
ED25519 key fingerprint is SHA256:jB7zKTS9wAIbBViMZH9eYSrIvyhKkd+nW5IXLWFkk/Q.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.111.168.44' (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 13:04:51 UTC 2024

System load:  0.58          Users logged in:      0
Usage of /:   51.6% of 7.57GB  IP address for eth0:    172.31.21.157
Memory usage: 26%
Swap usage:   0%           IP address for docker0: 172.17.0.1
Processes:    136          IP address for tunl0:   10.108.43.0

Expanded Security Maintenance for Infrastructure is not enabled.

8 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

89 additional security updates can be applied with ESM Infra.
Learn more about enabling ESM Infra service for Ubuntu 18.04 at
https://ubuntu.com/18-04

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-21-157:~$
```

```
ubuntu@ip-172-31-21-157:~$ sudo su
root@ip-172-31-21-157:/home/ubuntu# ps -ef | grep kubelet
root      5704  1  0 05:33 ?  00:04:00 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/kubelet.conf --config=/var/lib/kubelet/config.yaml --container-runtime=endpointunix:///var/run/containerd/containerd.sock --hostname-override=worker-0 --pod-infra-container-image=registry.k8s.io/pause:3.9
```

```
root@ip-172-31-21-157:/home/ubuntu# less /var/lib/kubelet/config.yaml |grep staticPodPath:
staticPodPath: /etc/kubernetes/manifests
root@ip-172-31-21-157:/home/ubuntu#
```

```

root@ip-172-31-21-157:/home/ubuntu# cd /etc/kubernetes/manifests
root@ip-172-31-21-157:/etc/kubernetes/manifests# ll
total 8
drwxr-xr-x 2 root root 4096 Feb 21 05:33 ../
drwxr-xr-x 4 root root 4096 Feb 21 05:33 ../
root@ip-172-31-21-157:/etc/kubernetes/manifests# touch static_pod.yaml
root@ip-172-31-21-157:/etc/kubernetes/manifests# vim static_pod.yaml
root@ip-172-31-21-157:/etc/kubernetes/manifests# cat static_pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: cloudethix-static
  labels:
    role: myrole
spec:
  containers:
  - name: web
    image: nginx
    ports:
    - name: web
      containerPort: 80
      protocol: TCP

```

```

root@ip-172-31-21-157:/etc/kubernetes/manifests# systemctl restart kubelet
root@ip-172-31-21-157:/etc/kubernetes/manifests# systemctl status kubelet
● kubelet.service - kubelet: The Kubernetes Node Agent
   Loaded: loaded (/lib/systemd/system/kubelet.service; enabled; vendor preset: enabled)
   Drop-In: /etc/systemd/system/kubelet.service.d
     └─10-kubeadm.conf
     Active: active (running) since Wed 2024-02-21 13:12:26 UTC; 8s ago
       Docs: https://kubernetes.io/docs/home/
     Main PID: 14714 (kubelet)
       Tasks: 9 (limit: 2342)
      CGroup: /system.slice/kubelet.service
              └─14714 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/kubelet.conf --config=/var/li
Feb 21 13:12:28 ip-172-31-21-157 kubelet[14714]: I0221 13:12:28.043171 14714 reconciler_common.go:258] "operationExecutor.VerifyControllerAttachedVolume s
Feb 21 13:12:28 ip-172-31-21-157 kubelet[14714]: I0221 13:12:28.043196 14714 reconciler_common.go:258] "operationExecutor.VerifyControllerAttachedVolume s
Feb 21 13:12:28 ip-172-31-21-157 kubelet[14714]: I0221 13:12:28.043298 14714 reconciler_common.go:258] "operationExecutor.VerifyControllerAttachedVolume s
Feb 21 13:12:28 ip-172-31-21-157 kubelet[14714]: I0221 13:12:28.043326 14714 reconciler_common.go:258] "operationExecutor.VerifyControllerAttachedVolume s
Feb 21 13:12:28 ip-172-31-21-157 kubelet[14714]: I0221 13:12:28.043384 14714 reconciler_common.go:258] "operationExecutor.VerifyControllerAttachedVolume s
Feb 21 13:12:28 ip-172-31-21-157 kubelet[14714]: I0221 13:12:28.043409 14714 reconciler_common.go:258] "operationExecutor.VerifyControllerAttachedVolume s
Feb 21 13:12:28 ip-172-31-21-157 kubelet[14714]: I0221 13:12:28.043472 14714 reconciler_common.go:258] "operationExecutor.VerifyControllerAttachedVolume s
Feb 21 13:12:28 ip-172-31-21-157 kubelet[14714]: I0221 13:12:28.043521 14714 reconciler_common.go:258] "operationExecutor.VerifyControllerAttachedVolume s
Feb 21 13:12:28 ip-172-31-21-157 kubelet[14714]: I0221 13:12:28.043558 14714 reconciler_common.go:258] "operationExecutor.VerifyControllerAttachedVolume s
Feb 21 13:12:28 ip-172-31-21-157 kubelet[14714]: I0221 13:12:28.044788 14714 reconciler_common.go:258] "operationExecutor.VerifyControllerAttachedVolume s

```

Que 7 →

- Install Kubectx & kubens in your k8s cluster.

SOLUTION:-

```

root@DESKTOP-VIDGD8F:Ques7# git clone https://github.com/ahmetb/kubectx.git ~/.kubectx
Cloning into '/root/.kubectx'...
remote: Enumerating objects: 1502, done.
remote: Counting objects: 100% (452/452), done.
remote: Compressing objects: 100% (98/98), done.
remote: Total 1502 (delta 390), reused 355 (delta 353), pack-reused 1050
Receiving objects: 100% (1502/1502), 912.88 KiB | 1.91 MiB/s, done.
Resolving deltas: 100% (876/876), done.

```

```

root@DESKTOP-VIDGD8F:Ques7# echo 'export PATH=$PATH:~/kubectx' >> ~/.bashrc
root@DESKTOP-VIDGD8F:Ques7# echo 'source ~/kubectx/completion/kubens.bash' >> ~/.bashrc
root@DESKTOP-VIDGD8F:Ques7# echo 'source ~/kubectx/completion/kubectx.bash' >> ~/.bashrc
root@DESKTOP-VIDGD8F:Ques7# source ~/.bashrc

```

```

root@DESKTOP-VIDGD8F:simple-webapp-docker# kubectx
kubernetes-admin@kubernetes
root@DESKTOP-VIDGD8F:simple-webapp-docker# █

```

```

root@DESKTOP-VIDGD8F:Ques7# kubens
default
kube-node-lease
kube-public
kube-system
root@DESKTOP-VIDGD8F:Ques7# █

```

Que 8 →

- Create 1 Public Docker Hub registry named flask_webapp_yourname.

The screenshot shows the Docker Hub interface for creating a new repository. The top navigation bar includes 'dockerhub', 'Explore', 'Repositories' (which is selected), and 'Organizations'. A search bar says 'Search Docker Hub' with a 'ctrl+K' keyboard shortcut. On the right are icons for help, grid, and settings. Below the bar, the path 'sanu28221 / Repositories / flask_webapp_saniya / General' is shown, along with a note 'Using 0 of 1 private repositories. [Get more](#)'. The 'General' tab is selected, with other tabs for 'Tags', 'Builds', 'Collaborators', 'Webhooks', and 'Settings'. A tooltip suggests adding a short description for the repository, noting it's used for indexing on Docker Hub and search engines. The repository name 'sanu28221/flask_webapp_saniya' is displayed with a copy icon. It was created less than a minute ago. A note states 'This repository does not have a description' with a pencil icon. To the right, under 'Docker commands', there's a button for 'Public View' and a command line box containing 'docker push sanu28221/flask_webapp_saniya:tagname'. Below this, sections for 'Tags' (empty) and 'Automated Builds' (with a note about connecting to GitHub or Bitbucket for automatic builds) are shown. An 'Upgrade' button is also present.

- Clone below repository on your system.
<https://github.com/mnumshad/simple-webapp-docker.git>
- Initialize a local repository & copy the code from above repo to your local repository in your working branch.

```

root@DESKTOP-VIDGD8F:Ques8# git clone https://github.com/mnumshad/simple-webapp-docker.git
Cloning into 'simple-webapp-docker'...
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 14 (delta 3), reused 2 (delta 2), pack-reused 7
Receiving objects: 100% (14/14), done.
Resolving deltas: 100% (3/3), done.
root@DESKTOP-VIDGD8F:Ques8# █

```

- 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-VIDGD8F:simple-webapp-docker# docker image push sanu28221/flask_webapp_saniya:flask_imagev1
The push refers to repository [docker.io/sanu28221/flask_webapp_saniya]
876bb5e552f5: Pushed
b17308435073: Pushed
c573c119b987: Pushed
28da0445c449: Mounted from library/ubuntu
flask_imagev1: digest: sha256:bc6de5ccf2d49b96c903105389818f36096e9de0d01a573d5e19d3c849058b4 size: 1160
root@DESKTOP-VIDGD8F:simple-webapp-docker#
```

sanu28221 / [Repositories](#) / [flask_webapp_saniya](#) / [General](#)

Using 0 of 1 private repositories. [Get more](#)

[General](#) Tags Builds Collaborators Webhooks Settings

i Add a short description for this repository [Update](#)

The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.

sanu28221/flask_webapp_saniya 

Updated less than a minute ago

This repository does not have a description 

Docker commands

To push a new tag to this repository:

```
docker push sanu28221/flask_webapp_saniya:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
 flask_imagev1		Image	--	a few seconds ago

[See all](#)

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#).

[Upgrade](#)

- 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-VIDGD8F:simple-webapp-docker# mkdir kube
root@DESKTOP-VIDGD8F:simple-webapp-docker# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 21 19:21 .
drwxrwxrwx 1 saniya saniya 512 Feb 21 19:12 ..
drwxrwxrwx 1 saniya saniya 512 Feb 21 19:12 .git
-rwxrwxrwx 1 saniya saniya 194 Feb 21 19:12 Dockerfile
-rwxrwxrwx 1 saniya saniya 229 Feb 21 19:12 app.py
drwxrwxrwx 1 saniya saniya 512 Feb 21 19:21 kube
root@DESKTOP-VIDGD8F:simple-webapp-docker# cd kube/
root@DESKTOP-VIDGD8F:kube# touch deployment.yaml
root@DESKTOP-VIDGD8F:kube# cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask-webapp
spec:
  selector:
    matchLabels:
      app: flask-webapp
  replicas: 3
  template:
    metadata:
      labels:
        app: flask-webapp
    spec:
      containers:
      - name: flask-webapp-cont
        image: sanu28221/flask_webapp_saniya:flask_imagev1
        ports:
        - containerPort: 8080
root@DESKTOP-VIDGD8F:kube# █

```

- Create one service.yaml file with type nodeport & select flask-webapp with port 8080 & targetPort 8080 with any nodePort between range 30000-32768.

```

total 0
drwxrwxrwx 1 saniya saniya 512 Feb 21 19:26 .
drwxrwxrwx 1 saniya saniya 512 Feb 21 19:21 ..
-rwxrwxrwx 1 saniya saniya 390 Feb 21 19:24 deployment.yaml
-rwxrwxrwx 1 saniya saniya 213 Feb 21 19:30 service.yaml
root@DESKTOP-VIDGD8F:kube# cat service.yaml
apiVersion: v1
kind: Service
metadata:
  name: flask-webapp
spec:
  selector:
    app: flask-webapp
  ports:
  - protocol: "TCP"
    port: 8080
    targetPort: 8080
    nodePort: 30011
  type: NodePort
root@DESKTOP-VIDGD8F:kube# █
```

- 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

```

root@DESKTOP-VIDGD8F:kube# kgp
NAME          READY   STATUS    RESTARTS   AGE
backend-7f5b7998b9-4dc9v   1/1     Running   0          167m
backend-7f5b7998b9-cv89b   1/1     Running   0          167m
backend-7f5b7998b9-l4xrk   1/1     Running   0          167m
cloudehix-static-worker-0  1/1     Running   0          49m
flask-webapp-74589f4f9c-b2sd4 1/1     Running   0          18s
flask-webapp-74589f4f9c-dfkl4 1/1     Running   0          18s
flask-webapp-74589f4f9c-jn682 1/1     Running   0          18s
frontend-85c84f8b8b-d47rk   1/1     Running   0          167m
master-nginx               1/1     Running   0          3h50m
prometheus-daemonset-6k7ws 1/1     Running   0          69m
prometheus-daemonset-wspxj  1/1     Running   0          69m
release-nginx              1/1     Running   0          3h40m
root@DESKTOP-VIDGD8F:kube# █
```

```

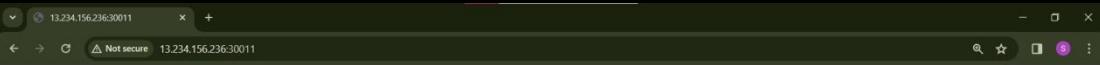
root@DESKTOP-VIDGD8F:kube# kgs
NAME        TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
cloudethix-clusterip ClusterIP  10.103.195.1  <none>        80/TCP        3h39m
flask-webapp  NodePort    10.97.227.72   <none>        8080:30011/TCP 5s
frontend      NodePort    10.102.226.115  <none>        80:30008/TCP  167m
hello         ClusterIP  10.102.180.156  <none>        80/TCP        167m
kubernetes    ClusterIP  10.96.0.1       <none>        443/TCP       8h
```

```
root@DESKTOP-VIDGD8F:simple-webapp-docker# cat app.py
import os
from flask import Flask
app = Flask(__name__)

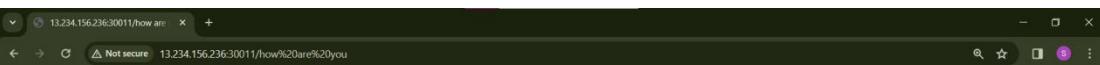
@app.route("/")
def main():
    return "Welcome!"

@app.route('/how are you')
def hello():
    return 'I am good, how about you?'

if __name__ == "__main__":
    app.run()
root@DESKTOP-VIDGD8F:simple-webapp-docker#
```



Welcome!



I am good, how about you?

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

```
root@DESKTOP-VIDGD8F:simple-webapp-docker# cat app.py
import os
from flask import Flask
app = Flask(__name__)

@app.route("/")
def main():
    return "Welcome!"

@app.route('/how are you')
def cloudethix():
    return 'Yes, I am cloudethix, and You !!!'

if __name__ == "__main__":
    app.run()
root@DESKTOP-VIDGD8F:simple-webapp-docker# vim app.py
root@DESKTOP-VIDGD8F:simple-webapp-docker# 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()
root@DESKTOP-VIDGD8F:simple-webapp-docker#
```

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

```
root@DESKTOP-VIDGDB8F:simple-webapp-docker# docker image push sanu28221/flask_webapp_saniya:flaskapp_imagev2
The push refers to repository [docker.io/sanu28221/flask_webapp_saniya]
94b94c9af9f3: Pushed
fa816442ea41: Layer already exists
7e9dfe8d5f4f: Layer already exists
28da0445c449: Layer already exists
flaskapp_imagev2: digest: sha256:6b24e87760bfa3ebd7849a5152b66fea64e946c531a228ea463467e9e03fe213 size: 1160
root@DESKTOP-VIDGDB8F:simple-webapp-docker#
```

- 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-VIDGDB8F:kube# kubectl set image deployment flask-webapp flask-webapp-cont=sanu28221/flask_webapp_saniya:flask_imagev1 --record
deployment.apps/flask-webapp image updated
root@DESKTOP-VIDGDB8F:kube# kubectl edit deployment flask-webapp
deployment.apps/flask-webapp edited
```

```
root@DESKTOP-VIDGDB8F:kube# kubectl set image deployment flask-webapp flask-webapp-cont=sanu28221/flask_webapp_saniya:flask_imagev2 --record
deployment.apps/flask-webapp image updated
root@DESKTOP-VIDGDB8F:kube# kubectl rollout history deployment flask-webapp
deployment.apps/flask-webapp
REVISION  CHANGE-CAUSE
1        kubectl set image deployment flask-webapp flask-webapp-cont=sanu28221/flask_webapp_saniya:flask_imagev1 --record=true
2        kubectl set image deployment flask-webapp flask-webapp-cont=sanu28221/flask_webapp_saniya:flask_imagev2 --record=true
root@DESKTOP-VIDGDB8F:kube#
```

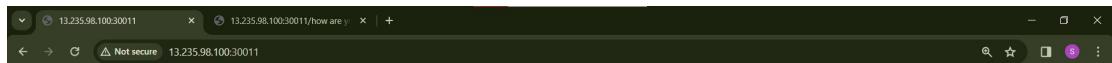
- Run the # kubectl rollout command to check status and history.

```
root@DESKTOP-VIDGDB8F:kube# kubectl rollout history deployment flask-webapp
deployment.apps/flask-webapp
REVISION  CHANGE-CAUSE
1        kubectl set image deployment flask-webapp flask-webapp-cont=sanu28221/flask_webapp_saniya:flask_imagev1 --record=true
2        kubectl set image deployment flask-webapp flask-webapp-cont=sanu28221/flask_webapp_saniya:flask_imagev2 --record=true
```

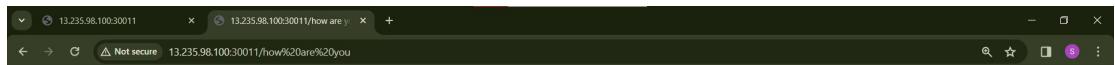
- Note:- Once above step 1 is done , run # kubectl rollout undo deployment command to rollback the change and then try a second way of rollout.

```
root@DESKTOP-VIDGDB8F:kube# kubectl rollout undo deployment flask-webapp --to-revision=1
deployment.apps/flask-webapp rolled back
root@DESKTOP-VIDGDB8F:kube# kubectl rollout history deployment flask-webapp
deployment.apps/flask-webapp
REVISION  CHANGE-CAUSE
2        kubectl set image deployment flask-webapp flask-webapp-cont=sanu28221/flask_webapp_saniya:flask_imagev2 --record=true
3        kubectl set image deployment flask-webapp flask-webapp-cont=sanu28221/flask_webapp_saniya:flask_imagev1 --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



Welcome!



I am good, how about you?

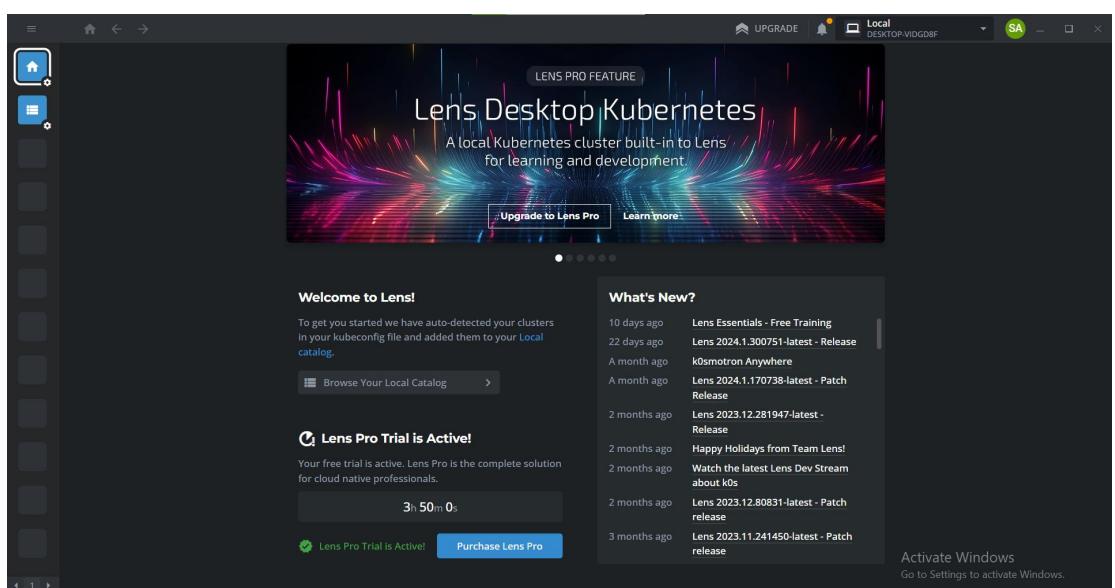


- Once done with all above steps , commit all the changes to the remote repository.
- Capture the snap and prepare a well formatted document.

SOLUTION:-

Que 9 →

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



Add Clusters from Kubeconfig

Clusters added here are **not** merged into the `~/.kube/config` file. Read more about adding clusters.

```

1  apiVersion: v1
2  clusters:
3  - cluster:
4    |   certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1J
5    |   server: https://13.235.98.100:6443
6    |   name: kubernetes
7  contexts:
8  - context:
9    |   cluster: kubernetes
10   |   user: kubernetes-admin
11   |   name: kubernetes-admin@kubernetes
12 current-context: kubernetes-admin@kubernetes
13 kind: Config
14 preferences: {}
15 users:
16 - name: kubernetes-admin
17   user:
18     |   client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUF
19     |   client-key-data: LS0tLS1CRUdJTiBSU0EgUFJJVkJURSBLRVktLS0tLQpNSUlFb3c

```

Add cluster

UPGRADE Local

Successfully added 1 new cluster(s)

Name	Source	Labels	Version	Distro	Status	⋮
kubernetes-admin@kubernetes	local				disconnected	⋮

- Create nginx Pod and Nodeport service. Check the Pod logs from Lens.

```

root@DESKTOP-VIDGD8F:Ques9# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 22 11:16 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 11:15 ..
-rwxrwxrwx 1 saniya saniya 335 Feb 22 11:20 nginx_pod.yaml
-rwxrwxrwx 1 saniya saniya 195 Feb 22 11:19 nodeport_svc.yaml

```

```
root@DESKTOP-VIDGD8F:Ques9# cat nginx_pod.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 1
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

```
root@DESKTOP-VIDGD8F:Ques9# cat nodeport_svc.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx
spec:
  selector:
    app: nginx
  ports:
  - protocol: "TCP"
    port: 80
    targetPort: 80
    nodePort: 32001
  type: NodePort
root@DESKTOP-VIDGD8F:Ques9#
```

```

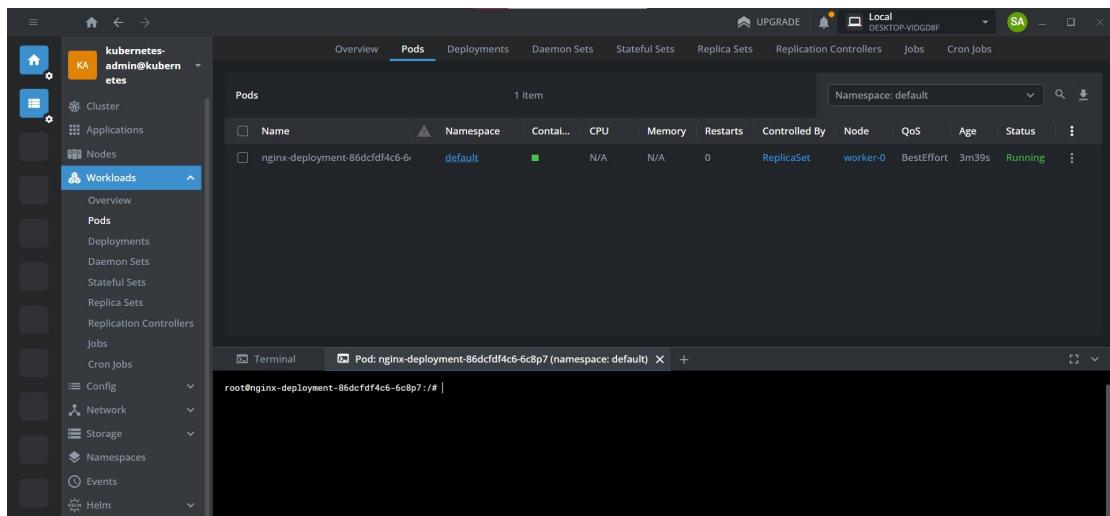
root@DESKTOP-VIDGD8F:Ques9# kcf .
deployment.apps/nginx-deployment created
service/nginx created
root@DESKTOP-VIDGD8F:Ques9# kgp
NAME                                READY   STATUS            RESTARTS   AGE
nginx-deployment-86dcfdf4c6-6c8p7   0/1    ContainerCreating   0          4s
root@DESKTOP-VIDGD8F:Ques9# kgs
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes  ClusterIP  10.96.0.1    <none>        443/TCP     45m
nginx     NodePort  10.98.74.32   <none>        80:32001/TCP  6s
root@DESKTOP-VIDGD8F:Ques9#

```

Name	Namespace	Contai...	CPU	Memory	Restarts	Controlled By	Node	QoS	Age	Status
nginx-deployment-86dcfdf4c6-6c8p7	default		N/A	N/A	0	ReplicaSet	worker-0	BestEffort	51s	Running

- Check the service from lens. Also login to the pod shell using the lens.

Name	Namespace	Type	Cluster IP	Ports	External IP	Selector	Age	Status
kubernetes	default	ClusterIP	10.96.0.1	443:6443/TCP	-	-	47m	Active
nginx	default	NodePort	10.98.74.32	80:32001/TCP	-	app=nginx	2m39s	Active



- Take snaps and delete the resources that you have just created.

SOLUTION:-

Que 10 →

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

The screenshot shows a Docker Hub repository page for the user 'sanu28221'. The repository name is 'cloudeithix_configmap_saniya'. The page includes sections for 'General' settings, Docker commands (with a placeholder for 'docker push'), and automated builds. The 'Tags' section is currently empty.

- 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 the working branch.

```
root@DESKTOP-VIDGD8F:Ques10# 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-VIDGD8F:Ques10#
```

```
root@DESKTOP-VIDGD8F:Ques10# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 22 11:30 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 11:29 ..
drwxrwxrwx 1 saniya saniya 512 Feb 22 11:29 .git
drwxrwxrwx 1 saniya saniya 512 Feb 22 11:30 docker-sample-nginx
root@DESKTOP-VIDGD8F:Ques10#
```

```
root@DESKTOP-VIDGD8F:docker-sample-nginx# ll
total 4
drwxrwxrwx 1 saniya saniya 512 Feb 22 11:30 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 11:30 ..
drwxrwxrwx 1 saniya saniya 512 Feb 22 11:30 .git
-rwxrwxrwx 1 saniya saniya 95 Feb 22 11:30 Dockerfile
-rwxrwxrwx 1 saniya saniya 1084 Feb 22 11:30 LICENSE
-rwxrwxrwx 1 saniya saniya 73 Feb 22 11:30 README.md
-rwxrwxrwx 1 saniya saniya 286 Feb 22 11:30 default.conf
-rwxrwxrwx 1 saniya saniya 103 Feb 22 11:30 index.html
root@DESKTOP-VIDGD8F:docker-sample-nginx#
```

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

```
root@DESKTOP-VIDGD8F:docker-sample-nginx# docker build -t sanu28221/cloudeithix_configmap_saniya:nginx_conf_image . --no-cache
[+] Building 36.0s (9/9) FINISHED                                            docker:default
=> [internal] load dockerignore                                         0.0s
=> => transferring context: 2B                                         0.0s
=> [internal] load build definition from Dockerfile                  0.0s
=> => transferring dockerfile: 132B                                         0.0s
=> [internal] load metadata for docker.io/library/nginx:alpine          35.5s
=> [auth] library/nginx:pull token for registry-1.docker.io           0.0s
=> => CACHED [1/3] FROM docker.io/library/nginx:alpine@sha256:6a2f0b28e45c4adea04ec207a251fd4a2d03ddc930f782af51e315ebc76e9a9
=> [internal] load build context                                         0.0s
=> => transferring context: 4698                                         0.0s
=> [2/3] COPY default.conf /etc/nginx/conf.d/                         0.0s
=> [3/3] COPY index.html /usr/share/nginx/html/                         0.0s
=> exporting image                                                       0.0s
=> => writing image sha256:526ead5ad80eeb48c38b37f336010ale8196b7elac28d2631fbaab390b5e5854
=> => naming to docker.io/sanu28221/cloudeithix_configmap_saniya:nginx_conf_image
root@DESKTOP-VIDGD8F:docker-sample-nginx#
```

```
root@DESKTOP-VIDGD8F:docker-sample-nginx# docker image push sanu28221/cloudeithix_configmap_saniya:nginx_conf_image
The push refers to repository [docker.io/sanu28221/cloudeithix_configmap_saniya]
3354bf5b1744: Pushed
58e0b4284a6f: Pushed
667a247707f0: Mounted from sanu28221/cloudeithix_release_nginx_saniya
d8527026595f: Mounted from sanu28221/cloudeithix_release_nginx_saniya
2593b08e5428: Mounted from sanu28221/cloudeithix_release_nginx_saniya
9909978d630d: Mounted from sanu28221/cloudeithix_release_nginx_saniya
c5140fc719dd: Mounted from sanu28221/cloudeithix_release_nginx_saniya
3137f8f0c641: Mounted from sanu28221/cloudeithix_release_nginx_saniya
718db50a47c0: Mounted from sanu28221/cloudeithix_release_nginx_saniya
aedc3bda2944: Mounted from sanu28221/cloudeithix_release_nginx_saniya
nginx_conf_image: digest: sha256:4aeaca76a83a8819a576753b526bef0456b75d87822ad1dd0e4438cc060d6a0b size: 2403
root@DESKTOP-VIDGD8F:docker-sample-nginx#
```

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

```
root@DESKTOP-VIDGD8F:kube# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 22 11:44 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 11:44 ..
-rwxrwxrwx 1 saniya saniya 411 Feb 22 11:47 deployment.yaml
root@DESKTOP-VIDGD8F:kube# cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend-webapp
spec:
  selector:
    matchLabels:
      app: frontend-webapp
  replicas: 3
  template:
    metadata:
      labels:
        app: frontend-webapp
    spec:
      containers:
      - name: frontend-webapp-cont
        image: sanu28221/cloudehix_configmap_saniya:nginx_conf_image
        ports:
        - containerPort: 80
root@DESKTOP-VIDGD8F:kube# █
```

- 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-VIDGD8F:kube# touch service.yaml
root@DESKTOP-VIDGD8F:kube# l
deployment.yaml*  service.yaml*
root@DESKTOP-VIDGD8F:kube# cat service.yaml
apiVersion: v1
kind: Service
metadata:
  name: frontend-webapp-svc
spec:
  selector:
    app: frontend-webapp
  ports:
  - protocol: "TCP"
    port: 80
    targetPort: 80
    nodePort: 32002
  type: NodePort
root@DESKTOP-VIDGD8F:kube# █

```

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

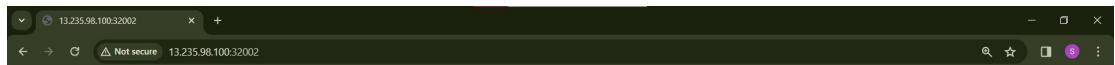
```

root@DESKTOP-VIDGD8F:kube# kcf .
deployment.apps/frontend-webapp created
service/service frontend-webapp-svc created

```

NAME	READY	STATUS	RESTARTS	AGE
frontend-webapp-649fc7f6f-46dnf	0/1	ContainerCreating	0	3s
frontend-webapp-649fc7f6f-hslxd	0/1	ContainerCreating	0	3s
frontend-webapp-649fc7f6f-kqln5	0/1	ContainerCreating	0	3s
nginx-deployment-86dcfdf4c6-6c8p7	1/1	Running	0	30m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
frontend-webapp-svc	NodePort	10.107.172.146	<none>	80:32002/TCP	5s
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	75m
nginx	NodePort	10.98.74.32	<none>	80:32001/TCP	30m



Host: frontend-webapp- 649fc7f6f-46dnf

Version: 1.1

- 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-VIDGD8F:kube# touch configmap.yaml
root@DESKTOP-VIDGD8F:kube# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 22 12:02 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 11:44 ..
-rwxrwxrwx 1 saniya saniya 0 Feb 22 12:02 configmap.yaml
-rwxrwxrwx 1 saniya saniya 411 Feb 22 11:47 deployment.yaml
-rwxrwxrwx 1 saniya saniya 219 Feb 22 11:52 service.yaml
root@DESKTOP-VIDGD8F:kube# cat configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: index-html-configmap
  namespace: default
data:
  index.html: |
    <html>
    <body>
    <h1> I am Cloudethix Team, Are you ?!! </h1>
    Version: 1.1
    </body>
    </html> root@DESKTOP-VIDGD8F:kube#
root@DESKTOP-VIDGD8F:kube#
```

```
root@DESKTOP-VIDGD8F:kube# k delete -f deployment.yaml
deployment.apps "frontend-webapp" deleted
root@DESKTOP-VIDGD8F:kube#
```

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

```
root@DESKTOP-VIDGD8F:kube# cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend-webapp
spec:
  selector:
    matchLabels:
      app: frontend-webapp
  replicas: 3
  template:
    metadata:
      labels:
        app: frontend-webapp
    spec:
      containers:
        - name: frontend-webapp-cont
          image: sanu28221/cloudehix_configmap_saniya:nginx_conf_image
          ports:
            - containerPort: 80
          volumeMounts:
            - name: config-volume
              mountPath: /usr/share/nginx/html/
      volumes:
        - name: config-volume
      configMap:
        name: index-html-configmap
root@DESKTOP-VIDGD8F:kube#
```

- 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-VIDGD8F:kube# ll
total 4
drwxrwxrwx 1 saniya saniya 512 Feb 22 12:02 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 11:44 ..
-rwxrwxrwx 1 saniya saniya 242 Feb 22 12:03 configmap.yaml
-rwxrwxrwx 1 saniya saniya 639 Feb 22 12:15 deployment.yaml
-rwxrwxrwx 1 saniya saniya 219 Feb 22 11:52 service.yaml
root@DESKTOP-VIDGD8F:kube# kubectl create -f configmap.yaml
configmap/index-html-configmap created
root@DESKTOP-VIDGD8F:kube# kubectl create -f deployment.yaml
deployment.apps/frontend-webapp created
root@DESKTOP-VIDGD8F:kube# kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
frontend-webapp-d5fc76677-hbhsd   1/1     Running   0          2s
frontend-webapp-d5fc76677-knwj7    1/1     Running   0          2s
frontend-webapp-d5fc76677-ln7v2    1/1     Running   0          2s
nginx-deployment-86dcfdf4c6-6c8p7  1/1     Running   0          53m
root@DESKTOP-VIDGD8F:kube# kubectl get configmap
NAME          DATA   AGE
index-html-configmap  1      23s
kube-root-ca.crt  1      98m
root@DESKTOP-VIDGD8F:kube# █

```



I am Cloudeithix Team, Are you ??!

Version: 1.1

Que 11 →

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

The screenshot shows a Docker Hub repository page. At the top, there are tabs for 'Explore', 'Repositories' (which is selected), 'Organizations', and a search bar. Below the header, the repository path is shown as `sanu28221 / Repositories / cloudeithix_multicontainer_saniya / General`. A message indicates 'Using 0 of 1 private repositories. [Get more](#)'. The main content area has a 'General' tab selected, showing a placeholder for a repository description and a 'Docker commands' section with a command box containing `docker push sanu28221/cloudeithix_multicontainer_saniya :tagname`. There are also sections for 'Tags' and 'Automated Builds'.

- Clone below repository on your system.
<https://github.com/janakiramm/Kubernetes-multi-container-pod.git>
- Initialize a local repository & copy the code from above repo to your local repository in any of your working branches.

```
root@DESKTOP-VIDGD8F:Ques11# 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 | 227.00 KiB/s, done.
Resolving deltas: 100% (21/21), done.
root@DESKTOP-VIDGD8F:Ques11#
```

```
root@DESKTOP-VIDGD8F:Ques11# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 22 12:24 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 12:22 ..
drwxrwxrwx 1 saniya saniya 512 Feb 22 12:24 .git
drwxrwxrwx 1 saniya saniya 512 Feb 22 12:24 Kubernetes-multi-container-pod
root@DESKTOP-VIDGD8F:Ques11#
```

- 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-VIDGD8F:Ques11# cd Kubernetes-multi-container-pod/
root@DESKTOP-VIDGD8F:Kubernetes-multi-container-pod# ll
total 120
drwxrwxrwx 1 saniya saniya 512 Feb 22 12:24 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 12:24 ..
drwxrwxrwx 1 saniya saniya 512 Feb 22 12:24 .git
-rwxrwxrwx 1 saniya saniya 9 Feb 22 12:24 .gitignore
drwxrwxrwx 1 saniya saniya 512 Feb 22 12:24 Build
drwxrwxrwx 1 saniya saniya 512 Feb 22 12:24 Deploy
-rwxrwxrwx 1 saniya saniya 2550 Feb 22 12:24 README.md
-rwxrwxrwx 1 saniya saniya 116003 Feb 22 12:24 multi-container-pod.png
root@DESKTOP-VIDGD8F:Kubernetes-multi-container-pod# cd Build/
root@DESKTOP-VIDGD8F:Build# ll
total 4
drwxrwxrwx 1 saniya saniya 512 Feb 22 12:24 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 12:24 ..
-rwxrwxrwx 1 saniya saniya 62 Feb 22 12:24 Dockerfile
-rwxrwxrwx 1 saniya saniya 1607 Feb 22 12:24 app.py
-rwxrwxrwx 1 saniya saniya 242 Feb 22 12:24 docker-compose.yml
-rwxrwxrwx 1 saniya saniya 24 Feb 22 12:24 requirements.txt

```

```

root@DESKTOP-VIDGD8F:Build# docker build -t sanu28221/cloudeithix_multicontainer_saniya:build_image . --no-cache
[+] Building 67.7s (10/10) FINISHED
--> [internal] load dockerignore
--> => transferring context: 2B
--> [internal] load build definition from Dockerfile
--> => transferring dockerfile: 90B
--> [internal] load metadata for docker.io/library/python:2.7-onbuild
--> => [auth] library/python:pull token for registry-1.docker.io
--> [+] FROM docker.io/library/python:2.7-onbuild@sha256:5af88e1d011b7e885e1813712d9f91be1a39e2ed092000fc53e0a0ce1333b
--> => resolve docker.io/library/python:2.7-onbuild@sha256:5af88e1d011b7e885e1813712d9f91be1a39e2ed092000fc53e0a0ce1333b
--> sha256:294d02320521b3eff91c0d7e59b1b3d527269807d80d6acf97900d1f5e3d9 2.22kB / 2.22kB
--> sha256:d56001f1509bfb8142f500518156f2d3b04f9e05854538b060498e2f7928d 54.25MB / 54.25MB
--> sha256:5af88e1d011b7e885e1813712d9f91be1a39e2ed092008fc53e0a0ce1333b 2.37kB / 2.37kB
--> sha256:3f2b400008a17001700e08a0d9e003706e9ea82b1e2fad0ff08040bbdd12 8.12kB / 8.12kB
--> sha256:68de23c3703d19122be597461c1a40bde1842aae7db728df20a9aabe11b 17.54MB / 17.54MB
--> sha256:68bae0745095220f6099d4aa703e4563c390e239a2d00b85bccc23a3ca3ac735 43.30MB / 43.30MB
--> sha256:e7f28bf935853e84197ae8340885e52b659922fice7e8672a37n62955550385fc7c7 131.10MB / 131.10MB
--> sha256:9c3de61682aa7de50535b5e9a9f3c37dfcc0173d6cc728369502d0c579400fc7c99 5.75MB / 5.75MB
--> sha256:50f10ddff177374ecfff540019a5f9d0c774ra561ea25b4d3b20a6e2ad34 14.59MB / 14.59MB
--> sha256:4073537c621dafa8bc2f0f6cfc1c73589587755822b40eaccabaa88f09f20f7e3c 1.78MB / 1.78MB
--> sha256:1106470fadic709cc13c5d6claaef04c2c43913a740c881bf47f0154e0166a512 3.66MB / 3.66MB
--> extracting sha256:68de23c3703d19122be597461c1a40bde1842aae7db728df20a9aabe11b
--> extracting sha256:68bae0745095220f6099d4aa703e4563c390e239a2d00b85bccc23a3ca3ac735
--> sha256:3delcccef08d9f0dc9899175d5e3735b0c228b6f22b31a211db5bd6009885f38a 130B / 130B
--> extracting sha256:e7f28bf935853e84197ae8340885e52b659922fice7e8672a37n62955550385fc7
--> extracting sha256:9c3de61682aa7de50535b5e9a9f3c37dfcc0173d6cc728369502d0c579400fc7c99
--> extracting sha256:4073537c621dafa8bc2f0f6cfc1c73589587755822b40eaccabaa88f09f20f7e3c
--> extracting sha256:3delcccef08d9f0dc989175d5e3735b0c228b6f22b31a211db5bd6009885f38a
--> extracting sha256:6809f0dc989175d5e3735b0c228b6f22b31a211db5bd6009885f38a
--> [internal] load build context
--> => transferring context: 2.18kB
--> [2/1] COPY requirements.txt /usr/src/app/
--> [3/1] RUN pip install --no-cache-dir -r requirements.txt
--> [4/1] COPY /usr/src/app
--> => exporting to image
--> => exporting layers
--> => writing image sha256:559a84301825f4eb4c81a525ecb21552cd82f68c9bc9091c2fc7eecef8df33
--> => naming to docker.io/samu28221/cloudeithix_multicontainer_saniya:build_image
root@DESKTOP-VIDGD8F:Build# 

```

```

root@DESKTOP-VIDGD8F:Build# docker image push sanu28221/cloudeithix_multicontainer_saniya:build_image
The push refers to repository [docker.io/samu28221/cloudeithix_multicontainer_saniya]
cf1106e74ea1: Pushed
ae9679540541: Pushed
c3e5724d0dee: Pushed
3e397f5b8357: Mounted from library/python
e257add70b4b: Mounted from library/python
ce7e990ce056: Mounted from library/python
633d23790c1d: Mounted from library/python
d071a18d9802: Mounted from library/python
8451f9fe0016: Mounted from library/python
858cd8541f7e: Mounted from library/python
a42d312a03bb: Mounted from library/python
dd1eb1fd7e08: Mounted from library/python
build_image: digest: sha256:022b422af9b5898edd301b287cdfdc4725c2d25162056fe9dcbe93c7c01bdb size: 2844
root@DESKTOP-VIDGD8F:Build# 

```

- Now go to the deploy directory and notice the files.

```
root@DESKTOP-VIDGD8F:Kubernetes-multi-container-pod# cd Deploy/
root@DESKTOP-VIDGD8F:Deploy# ll
total 1
drwxrwxrwx 1 saniya saniya 512 Feb 22 12:24 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 12:24 ..
-rwxrwxrwx 1 saniya saniya 325 Feb 22 12:24 db-pod.yml
-rwxrwxrwx 1 saniya saniya 203 Feb 22 12:24 db-svc.yml
-rwxrwxrwx 1 saniya saniya 467 Feb 22 12:24 web-pod-1.yml
-rwxrwxrwx 1 saniya saniya 467 Feb 22 12:24 web-pod-2.yml
-rwxrwxrwx 1 saniya saniya 644 Feb 22 12:24 web-rc.yml
-rwxrwxrwx 1 saniya saniya 218 Feb 22 12:24 web-svc.yml
root@DESKTOP-VIDGD8F:Deploy#
```

- 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.

```
root@DESKTOP-VIDGD8F:Deploy# cat web-pod-1.yml
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: sanu28221/cloudeithix_multicontainer_saniya:build_image
      env:
        - name: "REDIS_HOST"
          value: "localhost"
      ports:
        - containerPort: 5000
          name: http
          protocol: TCP
root@DESKTOP-VIDGD8F:Deploy#
```

```
root@DESKTOP-VIDGD8F:Deploy# kubectl create -f web-pod-1.yml
pod/web1 created
root@DESKTOP-VIDGD8F:Deploy#
```

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

```
root@DESKTOP-VIDGD8F:Deploy# cat 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
      nodePort: 32100
      protocol: TCP
root@DESKTOP-VIDGD8F:Deploy#
```

```
root@DESKTOP-VIDGD8F:Deploy# kubectl create -f web-svc.yml
service/web created
root@DESKTOP-VIDGD8F:Deploy#
```

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

```
root@DESKTOP-VIDGD8F:Deploy# cat db-pod.yml
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"
```

```
root@DESKTOP-VIDGD8F:Deploy#
```

```
root@DESKTOP-VIDGD8F:Deploy# kubectl create -f db-pod.yml
pod/mysql created
root@DESKTOP-VIDGD8F:Deploy#
```

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

```
root@DESKTOP-VIDGD8F:Deploy# cat db-svc.yml
apiVersion: v1
kind: Service
metadata:
  name: mysql
  labels:
    name: mysql
    app: demo
spec:
  ports:
    - port: 3306
      name: mysql
      targetPort: 3306
  selector:
    name: mysql
    app: demoroot@DESKTOP-VIDGD8F:Deploy#
```

```
root@DESKTOP-VIDGD8F:Deploy# kubectl create -f db-svc.yml
service/mysql created
root@DESKTOP-VIDGD8F:Deploy#
```

- Once above files are applied , Check that the Pods and Services are created using command line or lens.

NAME	READY	STATUS	RESTARTS	AGE
frontend-webapp-d5fc76677-hbhsd	1/1	Running	0	28m
frontend-webapp-d5fc76677-knwj7	1/1	Running	0	28m
frontend-webapp-d5fc76677-ln7v2	1/1	Running	0	28m
mysql	1/1	Running	0	2m14s
nginx-deployment-86dcfdf4c6-6c8p7	1/1	Running	0	82m
web1	1/2	CrashLoopBackOff	5 (2m8s ago)	5m35s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
frontend-webapp-svc	NodePort	10.107.172.146	<none>	80:32002/TCP	52m
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	127m
mysql	ClusterIP	10.96.161.123	<none>	3306/TCP	24s
nginx	NodePort	10.98.74.32	<none>	80:32001/TCP	82m
web	NodePort	10.101.60.48	<none>	80:32505/TCP	3m47s

- 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-VIDGD8F:Deploy# curl http://$NODE_IP:$NODE_PORT/init
curl: (3) URL using bad/illegal format or missing URL
```

```
root@DESKTOP-VIDGD8F:Deploy# export NODE_IP="13.235.98.100"
root@DESKTOP-VIDGD8F:Deploy# export NODE_PORT="32100"
root@DESKTOP-VIDGD8F:Deploy#
```

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

```
root@DESKTOP-VIDGD8F:Deploy# curl http://$NODE_IP:$NODE_PORT/init
DB Init done
root@DESKTOP-VIDGD8F:Deploy#
root@DESKTOP-VIDGD8F:Deploy#
```

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

```
root@DESKTOP-VIDGD8F:Deploy# curl -i -H "Content-Type: application/json" -X POST -d '{"uid": "1",
"user": "John Doe"}' http://$NODE_IP:$NODE_PORT/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 08:49:58 GMT

Added
root@DESKTOP-VIDGD8F:Deploy#
```

```
# curl -i -H "Content-Type: application/json" -X POST -d '{"uid": "2", "user": "Jane Doe"}' http://$NODE_IP:$NODE_PORT/users/add
```

```
root@DESKTOP-VIDGD8F:Deploy# curl -i -H "Content-Type: application/json" -X POST -d '{"uid": "2", "user": "Jane Doe"}' http://$NODE_IP:$NODE_PORT/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 08:50:34 GMT

Addedroot@DESKTOP-VIDGD8F:Deploy#
```

```
# curl -i -H "Content-Type: application/json" -X POST -d '{"uid": "3", "user": "Bill Colls"}' http://$NODE_IP:$NODE_PORT/users/add
```

```
root@DESKTOP-VIDGD8F:Deploy# curl -i -H "Content-Type: application/json" -X POST -d '{"uid": "3", "user": "Bill Colls"}' http://$NODE_IP:$NODE_PORT/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 08:50:59 GMT

Addedroot@DESKTOP-VIDGD8F:Deploy#
```

```
# curl -i -H "Content-Type: application/json" -X POST -d '{"uid": "4", "user": "Mike Taylor"}' http://$NODE_IP:$NODE_PORT/users/add
```

```
root@DESKTOP-VIDGD8F:Deploy# curl -i -H "Content-Type: application/json" -X POST -d '{"uid": "4", "user": "Mike Taylor"}' http://$NODE_IP:$NODE_PORT/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 08:51:33 GMT

Addedroot@DESKTOP-VIDGD8F:Deploy#
```

- Now access the data that we have added to database using below command.

```
# curl http://$NODE_IP:$NODE_PORT/users/1
```

```
Addedroot@DESKTOP-VIDGD8F:Depcurl http://$NODE_IP:$NODE_PORT/users/1
John Doeroot@DESKTOP-VIDGD8F:Deploy#
root@DESKTOP-VIDGD8F:Deploy#
```

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

```
root@DESKTOP-VIDGD8F:Deploy# curl http://$NODE_IP:$NODE_PORT/users/1
John Doe(c)root@DESKTOP-VIDGD8F:Deploy#
root@DESKTOP-VIDGD8F:Deploy#
```

- 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-VIDGD8F:Deploy# kubectl exec -it mysql -- /bin/sh
# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 5.7.25 MySQL Community Server (GPL)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

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 |
| USERDB          |
| mysql           |
| performance_schema |
| sys             |
+-----+
5 rows in set (0.00 sec)
```

```

mysql> use USERDB
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> select * from users;
+----+-----+
| ID | USER |
+----+-----+
| 1 | John Doe |
| 2 | Jane Doe |
| 2 | Jane Doe |
| 3 | Bill Colls |
| 4 | Mike Taylor |
+----+-----+
5 rows in set (0.00 sec)

mysql> exit
Bye
# exit
root@DESKTOP-VIDGD8F:Deploy#

```

- Prepare proper documentation in brief & write start to end flow. Refer below link if you face any issues.
<https://github.com/janakiramm/Kubernetes-multi-container-pod>

Que 12 →

- Create 1 Public Docker Hub registry named cloudeithix_Initcontainer_yourname.

The screenshot shows a Docker Hub repository page for the user 'sanu28221'. The repository name is 'cloudeithix_initcontainer_saniya'. The 'General' tab is selected. On the left, there's a section for adding a short description, which is currently empty. On the right, there are 'Docker commands' (with a placeholder for 'docker push'), a 'Public View' button, and sections for 'Tags' (empty) and 'Automated Builds' (also empty). The URL in the address bar is https://hub.docker.com/r/sanu28221/cloudeithix_initcontainer_saniya.

- Clone below repository on your system.
<https://github.com/janakiramm/simpleapp.git>

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

```
root@DESKTOP-VIDGD8F:Ques12# 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 | 158.00 KiB/s, done.
Resolving deltas: 100% (9/9), done.
root@DESKTOP-VIDGD8F:Ques12#
```

```
root@DESKTOP-VIDGD8F:Ques12# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 22 14:47 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 14:47 ..
drwxrwxrwx 1 saniya saniya 512 Feb 22 14:47 .git
drwxrwxrwx 1 saniya saniya 512 Feb 22 14:47 simpleapp
root@DESKTOP-VIDGD8F:Ques12#
```

- 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-VIDGD8F:simpleapp# docker build -t sanu28221/cloudeithix_initcontainer_saniya:simpleapp_image . --no-cache
[+] Building 49.1s (9/9) FINISHED
   => [internal] load .dockerignore
      0.1s
   => [internal] load context: 28
      0.0s
   => [internal] load build definition from Dockerfile
      0.1s
   => [internal] load metadata for docker.io/library/nginx:latest
      36.2s
   => [auth] library/nginx:pull token for registry-1.docker.io
      0.0s
   => [1/3] FROM docker.io/library/nginx@sha256:c26ae7472d624bafafad296e73cecc4f93f853088e6a9c13c0d52f6ca5865107
      11.4s
   => => resolve docker.io/library/nginx@sha256:c26ae7472d624bafafad296e73cecc4f93f853088e6a9c13c0d52f6ca5865107
      0.1s
   => => sha256:c3ea3340fe7111de02f171deebceh775ed1d9ee998f7fb472114dc1399cc6298 / 6298
      1.1s
   => => sha256:c26ae7472d624bafafad296e73cecc4f03f853088e6a9c13c0d52f6ca5865107
      0.8s
   => => sha256:05aa7306097caead48ea8213696b0f761cccd6802c53fc0a1a97a158301d71
      2.29kb / 2.29kb
   => => sha256:4e720693a3c1381245b53a5a1b4179631b3c4ff7d3df47ff301930a0ff3b17666a
      7.84kb / 7.84kb
   => => sha256:e1caac4eb9d2ec2faa3618e5992288321a92492ae5f5fe5f5e99e470895f771c56
      29.12mb / 29.12mb
   => => sha256:88fbf236f491ac87na5389d8ade2b0c9d24b9f526bd4e73311b5c1787cf4d49c
      41.39mb / 41.39mb
   => => sha256:cc1bb4345a3a849289c3e2471c896f37423ec1ef74766137b9de546498612
      9578 / 9578
   => => sha256:da8fa4352481b358fc69d40ee20d92da412hd2c700511560d0d5980330ff07e5
      3948 / 3948
   => => sha256:c7f809c9dab20387cd09e3c47121ef0eb531043cf0aca1a52ab659de3cc704
      1.21kb / 1.21kb
   => => sha256:10a509624cb69aa916902dc71c2b19ua78dccb99bf54du0951zeba54f70b8
      1.49kb / 1.49kb
   => => extracting sha256:e1caac4eb9d2ec24aa3618e5992288321a92492ae5f5fe5f5e99e470895f771c56
      1.75s
   => => extracting sha256:89f6f236f491ac87na5389d8ade2b0c9d24b9f526bd4e73311b5c1787cf4d49c
      1.28s
   => => extracting sha256:c1344e711f47111dee02f17deebce05725ed1d9ee998f7742114dc1399cc
      0.0s
   => => extracting sha256:cc1bb4345a3a849289c3e2471c896f37423ec1ef74766137b9de546498612
      0.0s
   => => extracting sha256:da8fa4352481b358fc69d40ee20d92da412hd2c700511560d0d5980330ff07e5
      0.0s
   => => extracting sha256:c7f809c9dab20387cd09e3c47121ef0eb531043cf0aca1a52ab659de3cc704
      0.0s
   => => extracting sha256:10a509624cb69aa9169042dc71c22b194a878dccb99bf54du0951zeba55f70b8
      0.0s
   => [internal] load build context
      0.1s
   => transferring context: 7858
      0.1s
   => [2/3] COPY wrapper.sh /
      0.0s
   => [3/3] COPY html /usr/share/nginx/html
      0.0s
   => exporting layers
      0.0s
   => => writing image sha256:a6a5319fd40b0405ee3f8f7de39d43f0f25033081363bd86e64a4f8be08d9766
      0.0s
   => => naming to docker.io/samu28221/cloudeithix_initcontainer_saniya:simpleapp_image
      0.0s
```

Activate Windows
Go to Settings to activate Windows 10

```
root@DESKTOP-VIDGD8F:simpleapp# docker image push samu28221/cloudeithix_initcontainer_saniya:simpleapp_image
The push refers to repository [docker.io/samu28221/cloudeithix_initcontainer_saniya]
adf0f51faf15: Pushed
c2532c700281: Pushed
61a7fb4dabcd: Mounted from library/nginx
bcc6856722b7: Mounted from library/nginx
188d128a188c: Mounted from library/nginx
7d52a4114c36: Mounted from library/nginx
3137f8f0c641: Mounted from samu28221/cloudeithix_configmap_saniya
84619992a45b: Mounted from library/nginx
ceb365432eec: Mounted from samu28221/example-voting-app
simpleapp_image: digest: sha256:0689f7d4b843d2ee6513b7376d994d0190dbae10ceba6ee6ba3aa131a1a180b8 size: 2192
root@DESKTOP-VIDGD8F:simpleapp#
```

- 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-VIDGD8F:kube# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 22 14:55 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 14:54 ..
-rwxrwxrwx 1 saniya saniya 0 Feb 22 14:55 deployment.yaml
root@DESKTOP-VIDGD8F:kube# cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: simpleapp-webapp
spec:
  selector:
    matchLabels:
      app: simpleapp-webapp
  replicas: 3
  template:
    metadata:
      labels:
        app: simpleapp-webapp
    spec:
      containers:
        - name: simpleapp-webapp-cont
          image: sanu28221/cloudeithix_initcontainer_saniya:simpleapp_image
          ports:
            - containerPort: 80
root@DESKTOP-VIDGD8F:kube#
```

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

```

root@DESKTOP-VIDGD8F:kube# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 22 14:58 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 14:54 ..
-rwxrwxrwx 1 saniya saniya 418 Feb 22 14:57 deployment.yaml
-rwxrwxrwx 1 saniya saniya 217 Feb 22 14:59 service.yaml
root@DESKTOP-VIDGD8F:kube# cat service.yaml
apiVersion: v1
kind: Service
metadata:
  name: simpleapp-webapp
spec:
  selector:
    app: simpleapp-webapp
  ports:
  - protocol: "TCP"
    port: 80
    targetPort: 80
    nodePort: 30015
  type: NodePort
root@DESKTOP-VIDGD8F:kube# █

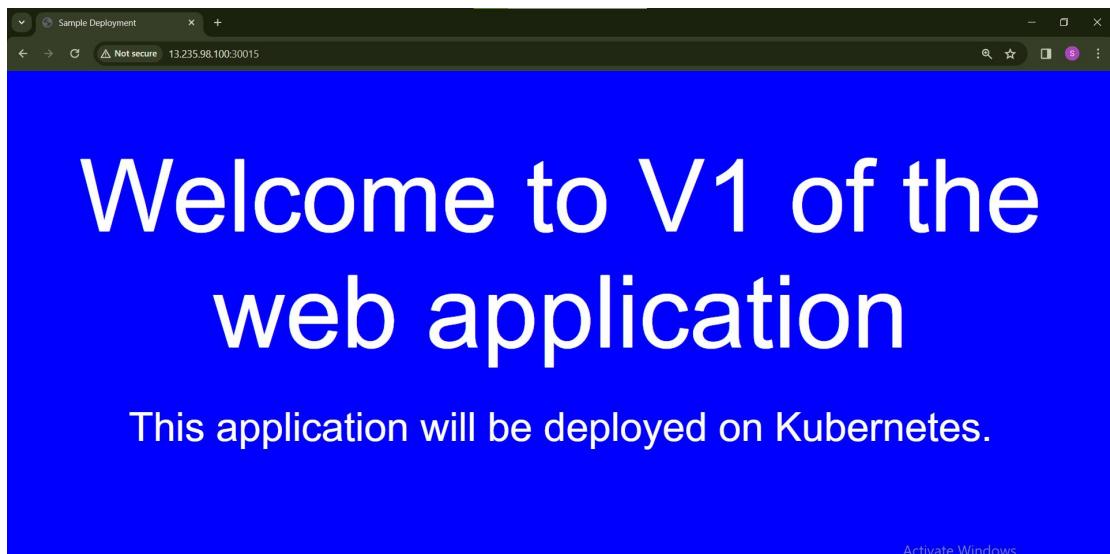
```

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

```

root@DESKTOP-VIDGD8F:kube# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 22 14:58 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 14:54 ..
-rwxrwxrwx 1 saniya saniya 418 Feb 22 14:57 deployment.yaml
-rwxrwxrwx 1 saniya saniya 217 Feb 22 15:04 service.yaml
root@DESKTOP-VIDGD8F:kube# kcf .
deployment.apps/simpleapp-webapp created
service/simpleapp-webapp created
root@DESKTOP-VIDGD8F:kube# kgp
NAME                  READY   STATUS    RESTARTS   AGE
simpleapp-webapp-569ff88f5d-bnxks  1/1     Running   0          20s
simpleapp-webapp-569ff88f5d-q7cqv  1/1     Running   0          20s
simpleapp-webapp-569ff88f5d-t2f8d  1/1     Running   0          20s
root@DESKTOP-VIDGD8F:kube# kgs
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
kubernetes     ClusterIP   10.96.0.1      <none>         443/TCP       5h13m
simpleapp-webapp   NodePort    10.111.116.75  <none>         80:30015/TCP  21m
root@DESKTOP-VIDGD8F:kube# █

```



- Then delete the deployment that you have just created.

```
root@DESKTOP-VIDGD8F:kube# kubectl delete -f deployment.yaml
deployment.apps "simpleapp-webapp" deleted
root@DESKTOP-VIDGD8F:kube#
```

- Update the deployment.yaml file and add volumeMounts with mountPath /usr/share/nginx/html from emptyDir: {} volume.

```
root@DESKTOP-VIDGD8F:kube# cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: simpleapp-webapp
spec:
  selector:
    matchLabels:
      app: simpleapp-webapp
  replicas: 3
  template:
    metadata:
      labels:
        app: simpleapp-webapp
    spec:
      containers:
        - name: simpleapp-webapp-cont
          image: sanu28221/cloudeithix_initcontainer_saniya:simpleapp_image
          ports:
            - containerPort: 80
          volumeMounts:
            - name: volume
              mountPath: /usr/share/nginx/html/
  volumes:
    - name: volume
      emptyDir: {}root@DESKTOP-VIDGD8F:kube#
```

- Once above changes are added, add initContainers block with below parameters. Also add volumeMounts for Init Container with mountPath "/work-dir" from emptyDir: {} volume.

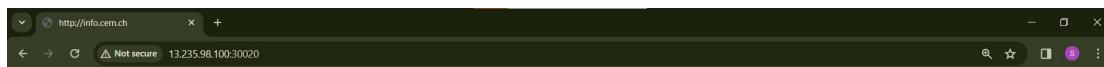
```
initContainers:
- name: install
  image: busybox:1.28
  command:
  - wget
  - "-O"
  - "/work-dir/index.html"
  - http://info.cern.ch
volumeMounts:
- name: workdir
  mountPath: "/work-dir"
```

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

```
root@DESKTOP-VIDGD8F:kube# cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: simpleapp-webapp
spec:
  selector:
    matchLabels:
      app: simpleapp-webapp
  replicas: 3
  template:
    metadata:
      labels:
        app: simpleapp-webapp
    spec:
      containers:
        - name: simpleapp-webapp-cont
          image: 'sanu28221/cloudeithix_initcontainer_saniya:simpleapp_image'
          ports:
            - containerPort: 80
          volumeMounts:
            - name: html-volume
              mountPath: /usr/share/nginx/html
      initContainers:
        - name: install
          image: busybox:1.28
          command:
            - wget
            - "-O"
            - "/work-dir/index.html"
            - http://info.cern.ch
          volumeMounts:
            - name: workdir
              mountPath: "/work-dir"
      volumes:
        - name: html-volume
          emptyDir: {}
        - name: workdir
          emptyDir: {}
```

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

```
root@DESKTOP-VIDGD8F:kube# kubectl apply -f deployment.yaml
deployment.apps/simpleapp-webapp created
root@DESKTOP-VIDGD8F:kube#
```



http://info.cern.ch - home of the first website

From here you can:

- [Browse the first website](#)
- [Browse the first website using the line-mode browser simulator](#)
- [Learn about the birth of the web](#)
- [Learn about CERN, the physics laboratory where the web was born](#)

Activate Windows
Go to Settings to activate Windows.

- Prepare a well formatted document and write your understanding step by step.

SOLUTION:-

Que 13 →

- Create 1 Public Docker Hub registry named cloudethix_hpa_yourname.

The screenshot shows the Docker Hub interface for a repository named "cloudethix_hpa_saniya" owned by "sanu28221". The "General" tab is active. A note says "Add a short description for this repository" with an "Update" button. The repository was created less than a minute ago. The Docker commands section includes a "Public View" button and a command line input field: "docker push sanu28221/cloudethix_hpa_saniya:tagname". The "Tags" section is currently empty. The "Automated Builds" section notes that manually pushing images to Hub? is possible with GitHub or Bitbucket, and it's available with Pro, Team and Business subscriptions. An "Upgrade" button is also present.

- Clone below repository on your system.

<https://github.com/vivekamin/kubernetes-hpa-example.git>

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

```
root@DESKTOP-VIDGD8F:Ques13# 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-VIDGD8F:Ques13# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 22 17:25 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 17:24 ..
drwxrwxrwx 1 saniya saniya 512 Feb 22 17:25 .git
drwxrwxrwx 1 saniya saniya 512 Feb 22 17:25 kubernetes-hpa-example
root@DESKTOP-VIDGD8F:Ques13#
```

- 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-VIDGD8F:kubernetes-hpa-example# ll
total 4
drwxrwxrwx 1 saniya saniya 512 Feb 22 17:25 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 17:25 ..
drwxrwxrwx 1 saniya saniya 512 Feb 22 17:25 .git
-rwxrwxrwx 1 saniya saniya 127 Feb 22 17:25 Dockerfile
-rwxrwxrwx 1 saniya saniya 2788 Feb 22 17:25 README.md
drwxrwxrwx 1 saniya saniya 512 Feb 22 17:25 k8s
-rwxrwxrwx 1 saniya saniya 272 Feb 22 17:25 package.json
drwxrwxrwx 1 saniya saniya 512 Feb 22 17:25 src
```

```
root@DESKTOP-VIDGD8F:kubernetes-hpa-example# docker build -t sanu28221/cloudeithix_hpa_saniya:hpa_image . --no-cache
[+] Building 38.8s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load .dockerignore
=> [internal] load context: 2B
=> [internal] load metadata for docker.io/library/node:8.12.0-alpine
=> [auth] library/node:pull token for registry-1.docker.io
=> [1/5] FROM docker.io/library/node:8.12.0-alpine@sha256:d75742c5fd41261113ed4786f961a21238db8048c825a5126ada373c361f46e
=> => resolve docker.io/library/node:8.12.0-alpine@sha256:d75742c5fd41261113ed4786f961a21238db8048c825a5126ada373c361f46e
=> => sha256 d75742c5fd41261113ed4786f961a21238db8048c825a5126ada373c361f46e 2.03kB / 2.03kB
=> => sha256_81abbd8de10e0b6e55bcfa103b3c2e1e2d167c2cd1391a0d222f7c5e710_9518 / 9518
=> => sha256_dFU0h65da02a3ac8e7df87a5024888aef969952d9941c93346352f673bb135e27 5.09kB / 5.09kB
=> => sha256_0fe2ad4980c7d4a4fc95858eb9811898baec8c1eHd282ah1c3569befff9bde 2.21MB / 2.21MB
=> => sha256_eeb7d76f00071e809d66e8491376534c89deaa0b607501e1f76b694912ad646 18.82MB / 18.82MB
=> => sha256_e35fb8fcc25962e49894d767da2u79ec9ce1b503ef2fe0e800bad07bc52438319 1.08MB / 1.08MB
=> => extracting sha256_4f7eade49890c2da4fc95858eb901480baec8c1e0bd282ab1c3569befff9bde
=> => extracting sha256_web7d76f4471e889460e84491576514c80deea0b607581e14f476b6040124d46
=> [internal] load build context
=> [internal] transfer context: 38.33kB
=> [2/5] RUN mkdir -p /usr/src/app
=> [3/5] WORKDIR /usr/src/app
=> [4/5] COPY . /usr/src/app
=> [5/5] RUN npm install
=> exporting to image
=> => exporting layers
=> => writing image sha256_8d107c1fc891e37fd44483f259b3e348bac25a70391fa1f1858817af0ab95248
=> => naming to docker.io/sanu28221/cloudeithix_hpa_saniya:hpa_image
root@DESKTOP-VIDGD8F:kubernetes-hpa-example#
```

Activate Windows
Go to Settings to activate Windows.

```
root@DESKTOP-VIDGD8F:k8s# docker image push sanu28221/cloudehix_hpa_saniya:hpa_image
The push refers to repository [docker.io/sanu28221/cloudehix_hpa_saniya]
cfdc942db280: Pushed
59677765333b: Pushed
5f70bf18a086: Mounted from sanu28221/haproxy_test
1d469a49de79: Pushed
8b59e4cead98: Mounted from library/node
7aa09d2ca0a3: Mounted from library/node
df64d3292fd6: Mounted from library/node
hpa_image: digest: sha256:572356db8904fc7681b3093a43fba533d4d709c850414b8c1511e449b52b835c size: 1780
root@DESKTOP-VIDGD8F:k8s#
```

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

```
root@DESKTOP-VIDGD8F:k8s# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 22 17:25 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 17:25 ..
-rwxrwxrwx 1 saniya saniya 465 Feb 22 17:31 deployment.yml
-rwxrwxrwx 1 saniya saniya 291 Feb 22 17:25 hpa.yml
-rwxrwxrwx 1 saniya saniya 214 Feb 22 17:25 service.yml
root@DESKTOP-VIDGD8F:k8s# cat deployment.yml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: node-example
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: node-example
    spec:
      containers:
        - name: node-example
          image: sanu28221/cloudehix_hpa_saniya:hpa_image
          imagePullPolicy: Always
          ports:
            - containerPort: 3000
          resources:
            limits:
              cpu: "0.5"
            requests:
              cpu: "0.25"root@DESKTOP-VIDGD8F:k8s#
```

```
root@DESKTOP-VIDGD8F:k8s# ll
total 1
drwxrwxrwx 1 saniya saniya 512 Feb 22 17:25 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 17:25 ..
-rwxrwxrwx 1 saniya saniya 538 Feb 22 17:45 deployment.yml
-rwxrwxrwx 1 saniya saniya 291 Feb 22 17:25 hpa.yml
-rwxrwxrwx 1 saniya saniya 214 Feb 22 17:25 service.yml
root@DESKTOP-VIDGD8F:k8s# kubectl apply -f deployment.yml
deployment.apps/node-example created
root@DESKTOP-VIDGD8F:k8s# █
```

```
root@DESKTOP-VIDGD8F:k8s# kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
node-example-c6bf7bdc4-58lnt   1/1     Running   0          10m
simpleapp-webapp-7d574bf6c5-bsqnd   1/1     Running   0          6m53s
root@DESKTOP-VIDGD8F:k8s#
```

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

```
root@DESKTOP-VIDGD8F:k8s# cat service.yml
apiVersion: v1
kind: Service
metadata:
  name: node-example
  labels:
    app: node-example
spec:
  selector:
    app: node-example
  ports:
  - port: 3000
    protocol: TCP
    nodePort: 30001
  type: NodePort
root@DESKTOP-VIDGD8F:k8s# █
```

```
root@DESKTOP-VIDGD8F:k8s# kubectl apply -f service.yml
service/node-example created
root@DESKTOP-VIDGD8F:k8s# █
```

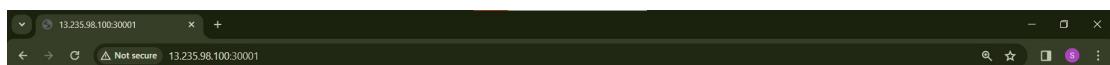
```
root@DESKTOP-VIDGD8F:k8s# kgs
NAME          TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes    ClusterIP   10.96.0.1      <none>         443/TCP      7h21m
node-example   NodePort    10.99.78.38    <none>         3000:30001/TCP 4m55s
simpleapp-webapp-service NodePort    10.110.105.22   <none>         80:30016/TCP   65m
root@DESKTOP-VIDGD8F:k8s#
```

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

```
root@DESKTOP-VIDGD8F:k8s# ll
total 1
drwxrwxrwx 1 saniya saniya 512 Feb 22 17:25 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 17:25 ..
-rwxrwxrwx 1 saniya saniya 538 Feb 22 17:45 deployment.yml
-rwxrwxrwx 1 saniya saniya 295 Feb 22 18:00 hpa.yml
-rwxrwxrwx 1 saniya saniya 210 Feb 22 17:54 service.yml
root@DESKTOP-VIDGD8F:k8s# cat hpa.yml
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  annotations:
    name: node-example
    namespace: default
spec:
  maxReplicas: 4
  minReplicas: 1
  scaleTargetRef:
    apiVersion: extensions/v1
    kind: Deployment
    name: node-example
  targetCPUUtilizationPercentage: 1
root@DESKTOP-VIDGD8F:k8s#
```

```
root@DESKTOP-VIDGD8F:k8s# kubectl create -f hpa.yml
horizontalpodautoscaler.autoscaling/node-example-hpa created
root@DESKTOP-VIDGD8F:k8s#
```

- Open the browser, and access the webpage.



Hello World!!

- 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 -O  
http://NODE_PORT_SERVICE_NAME; done"
```

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

```
root@DESKTOP-VIDGD8F:k8s# kubectl get hpa
NAME           REFERENCE          TARGETS      MINPODS   MAXPODS   REPLICAS   AGE
node-example-hpa   Deployment/node-example   <unknown>/80%   1         4          1          50m
root@DESKTOP-VIDGD8F:k8s#
```

- Take the snap , prepare a well formatted doc and write your understanding.

SOLUTION:-

Que 14 →

- Create 1 Public Docker Hub registry named cloudethix cronjob yourname.

The screenshot shows a Docker Hub repository page for the user `sanu28221`. The repository name is `cloudethix_cronjob_saniya`. The `General` tab is selected. A note says "Add a short description for this repository" and "The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results." There is a "Public View" button. Under "Docker commands", there is a box containing the command `docker push sanu28221/cloudethix_cronjob_saniya:tagname`. The "Tags" section notes that the repository is empty. The "Automated Builds" section mentions that manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating. An "Upgrade" button is present.

- Initialize a local repository & copy below code (three files) to your local repository in any of your working branch.

```
root@DESKTOP-VIDGD8F:Ques14# touch helloworld.py
root@DESKTOP-VIDGD8F:Ques14# touch Dockerfile
root@DESKTOP-VIDGD8F:Ques14# touch pythoncronjob.yml
root@DESKTOP-VIDGD8F:Ques14# ll
total 0
drwxrwxrwx 1 saniya saniya 512 Feb 22 19:11 .
drwxrwxrwx 1 saniya saniya 512 Feb 22 19:09 ..
-rwxrwxrwx 1 saniya saniya 0 Feb 22 19:11 Dockerfile
-rwxrwxrwx 1 saniya saniya 0 Feb 22 19:11 helloworld.py
-rwxrwxrwx 1 saniya saniya 0 Feb 22 19:11 pythoncronjob.yml
root@DESKTOP-VIDGD8F:Ques14#
```

```
root@DESKTOP-VIDGD8F:Ques14# cat 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-VIDGD8F:Ques14#
```

```
root@DESKTOP-VIDGD8F:Ques14# cat helloworld.py
#!/usr/local/bin/python3
import datetime
x = datetime.datetime.now()
print("Welcome to the Cloudethix World")
print("Today is")
print(x)root@DESKTOP-VIDGD8F:Ques14#
```

```
root@DESKTOP-VIDGD8F:Ques14# cat pythoncronjob.yml
apiVersion: batch/v1
kind: CronJob
metadata:
  name: python-helloworld
spec:
  schedule: "*/1 * * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
            - name: python-helloworld
              image: python-helloworld
              command: ["/app/helloworld.py"]
            restartPolicy: OnFailure
root@DESKTOP-VIDGD8F:Ques14#
```

- Once code is copied, build the docker image from Dockerfile , add meaningful tags and then push the docker image to Docker hub registry.

```
root@DESKTOP-VIDGD8F:Ques14# docker build -t sanu28221/cloudethix_cronjob_saniya:python_image . --no-cache
[+] Building 20.0s (11/11) docker:default
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 306B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/python:3.7-alpine
=> [auth] library/python:pull token for registry1.docker.io
=> [1/5] FROM docker.io/library/python:3.7-alpine@sha256:f3d31c8677d03f0b3c724046077f229a6ce9d3ac430f5c08cd7df00292048c3
=> => resolve docker.io/library/python:3.7-alpine@sha256:f3d31c8677d083f0b3c724046077f229a6ce9d3ac430f5c08cd7df00292048c3
=> => sha256:ibac8ae77e4af0b66bb2a75115616a20e025e0451eeed05d94a4cfc4523e58a 0.87kB
=> => sha256:9b526aa774ef0126ad0fe9e9a95764c5fc37fd099ab9e97821e7b47758d82b0f6fa 3.40MB / 3.40MB
=> => sha256:9875af95546db7816baa761b7fa205edc0d0c153cd89356c1512e551c12b2d5c 622.29kB / 622.29kB
=> => sha256:4819c95474fc4a04767c9129b02238ebcce0bbc682280cb671379bc1f708a12b55 10.94MB / 10.94MB
=> => sha256:f3d31c6774083fb3c724046077f229a6ce9d3ac430f5c08cd7df00292048c3 1.65kB / 1.65kB
=> => sha256:e5da3se9bb640d12b98fa609087f112fe1e8305389db15c22a8a51 1.37kB / 1.37kB
=> => sha256:148762f75a1f92cc9857e9c88b8f9d5aacc1e990sec47a7488025b2dd5c3b7a 240B / 240B
=> => extracting sha256:9b526aa774ef0126ad0fe9e9a95704c5fc37fd099ab9e97821e7b47758d82b0f6fa
=> => sha256:ea1518237b4753b3fe48ee773d77651704178d9baa72aa5912e13a992cfac63 2.85MB / 2.85MB
=> => extracting sha256:9875af95546db7816baa761b7fa205edc0d0c153cd89356c1512e551c12b2d5c
=> => extracting sha256:4819c95424fc4a04767c9129b02238ebcce0bbc682280cb671379bc1f8a12b55
=> => extracting sha256:148762f75a1f92cc9857e9c88b8f9d5aacc1e990sec47a7488025b2dd5c3b7a
=> => sha256:ea1518237b4753b3fe48ee773d77651704178d9baa72aa5912e13a992cfac63
=> [internal] load build context
=> => transferring context: 183B
=> [2/5] RUN addgroup -s appgroup && adduser -G appgroup
=> [3/5] WORKDIR /app
=> [4/5] COPY helloworld.py /app/helloworld.py
=> [5/5] RUN chmod +x /app/helloworld.py
=> => exporting image sha256:857940658be5e17793f6143e3a1c14a90e82aacd8256640df0f5281d6fdf8bcd
=> => naming to docker.io/sanu28221/cloudethix_cronjob_saniya:python_image
Activate Windows 0.0s
Go to Settings to activate Windows.
root@DESKTOP-VIDGD8F:Ques14#
```

```

root@DESKTOP-VIDGD8F:Ques14# docker image push sanu28221/cloudethix_cronjob_saniya:python_image
The push refers to repository [docker.io/sanu28221/cloudethix_cronjob_saniya]
089d0a3e4969: Pushed
438a13a2801c: Pushed
f68d7866ba10: Pushed
c73bf84db3b0: Pushed
ae2ed3079163: Mounted from library/python
aa3a591fc84e: Mounted from library/python
7f29b11ef9dd: Mounted from library/python
a1c2f058ec5f: Mounted from library/python
cc2447e1835a: Mounted from library/python
python_image: digest: sha256:627f7442c1cd1649053a667b859f94b79d4eb981cbc91c5573319e9637e26c76 size: 2196
root@DESKTOP-VIDGD8F:Ques14#

```

The screenshot shows the Docker Hub interface for the repository 'sanu28221/cloudethix_cronjob_saniya'. The 'General' tab is active. A placeholder for a short description is present. The Docker commands section contains a command to push a new tag. The 'Tags' section lists one tag: 'python_image'. The 'Automated Builds' section provides information on manually pushing images to Hub.

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

```

root@DESKTOP-VIDGD8F:Ques14# cat pythoncronjob.yml
apiVersion: batch/v1
kind: CronJob
metadata:
  name: python-helloworld
spec:
  schedule: "*/1 * * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
            - name: python-helloworld
              image: sanu28221/cloudethix_cronjob_saniya:python_image
              command: ["/app/helloworld.py"]
              restartPolicy: OnFailure
root@DESKTOP-VIDGD8F:Ques14#

```

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

```
root@DESKTOP-VIDGD8F:Ques14# kubectl create -f pythoncronjob.yml
cronjob.batch/python-helloworld created
root@DESKTOP-VIDGD8F:Ques14#
```

```
root@DESKTOP-VIDGD8F:Ques14# kubectl get cronjobs
NAME          SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
python-helloworld  */1 * * * *  False     1        24s      84s
root@DESKTOP-VIDGD8F:Ques14#
```

- Check the Job name which is created by cronjob from command line or lens.

```
root@DESKTOP-VIDGD8F:Ques14# kubectl get jobs
NAME                  COMPLETIONS   DURATION   AGE
python-helloworld-28477786  0/1          6m14s    6m14s
python-helloworld-28477787  0/1          5m14s    5m14s
python-helloworld-28477788  0/1          4m14s    4m14s
python-helloworld-28477789  0/1          3m14s    3m14s
python-helloworld-28477790  0/1          2m14s    2m14s
python-helloworld-28477791  0/1           74s      74s
python-helloworld-28477792  0/1           14s      14s
root@DESKTOP-VIDGD8F:Ques14#
```

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

```
root@DESKTOP-VIDGD8F:Ques14# kgp
NAME          READY   STATUS      RESTARTS   AGE
python-helloworld-28477786-nkkxf  0/1   CrashLoopBackOff  5 (2m13s ago)  5m9s
python-helloworld-28477787-r59rb  0/1   CrashLoopBackOff  5 (76s ago)   4m9s
python-helloworld-28477788-wk768  0/1   Error          5 (93s ago)   3m9s
python-helloworld-28477789-qhxq2  0/1   CrashLoopBackOff  4 (47s ago)   2m9s
python-helloworld-28477790-dsqft  0/1   CrashLoopBackOff  3 (28s ago)   69s
python-helloworld-28477791-69fff5 0/1   CrashLoopBackOff  1 (7s ago)    9s
root@DESKTOP-VIDGD8F:Ques14#
```

- Prepare well formatted documents and write your understanding.

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

# vim 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"]

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