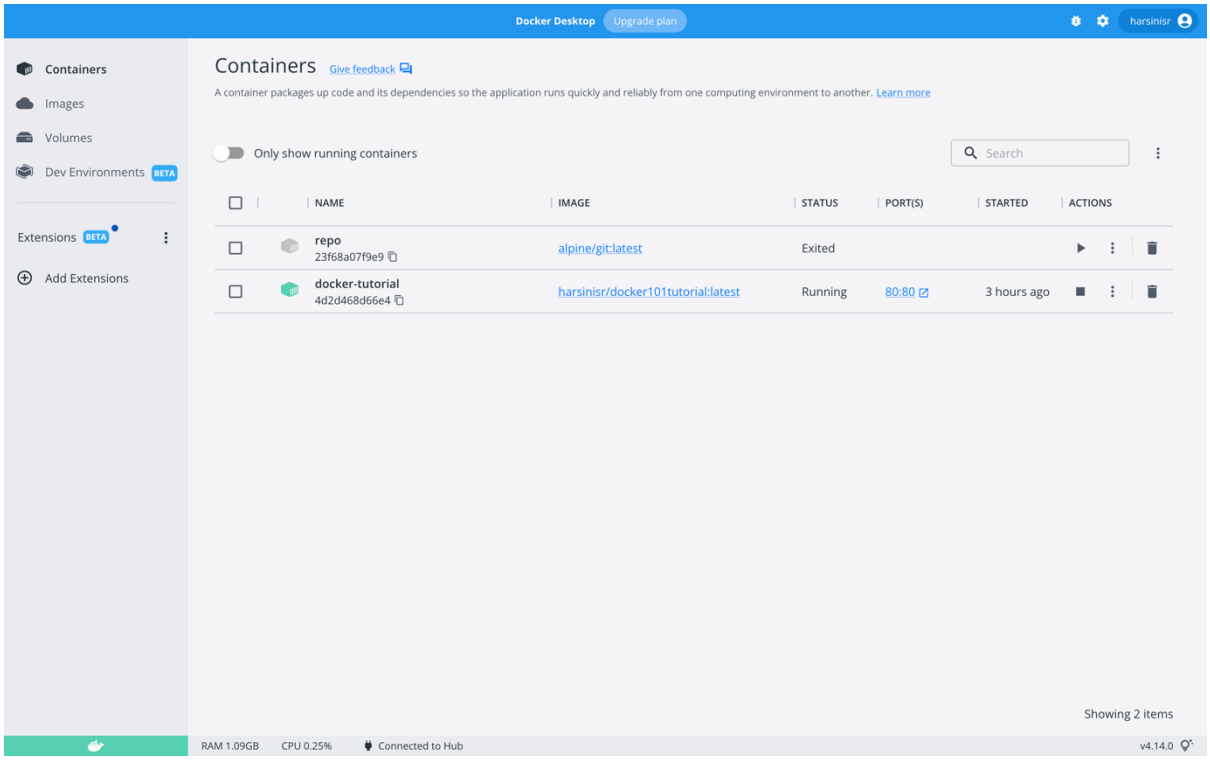
Assignment 4
Kubernetes / Docker

Student Name	Harsini S.R.
Student Roll Number	111519205011

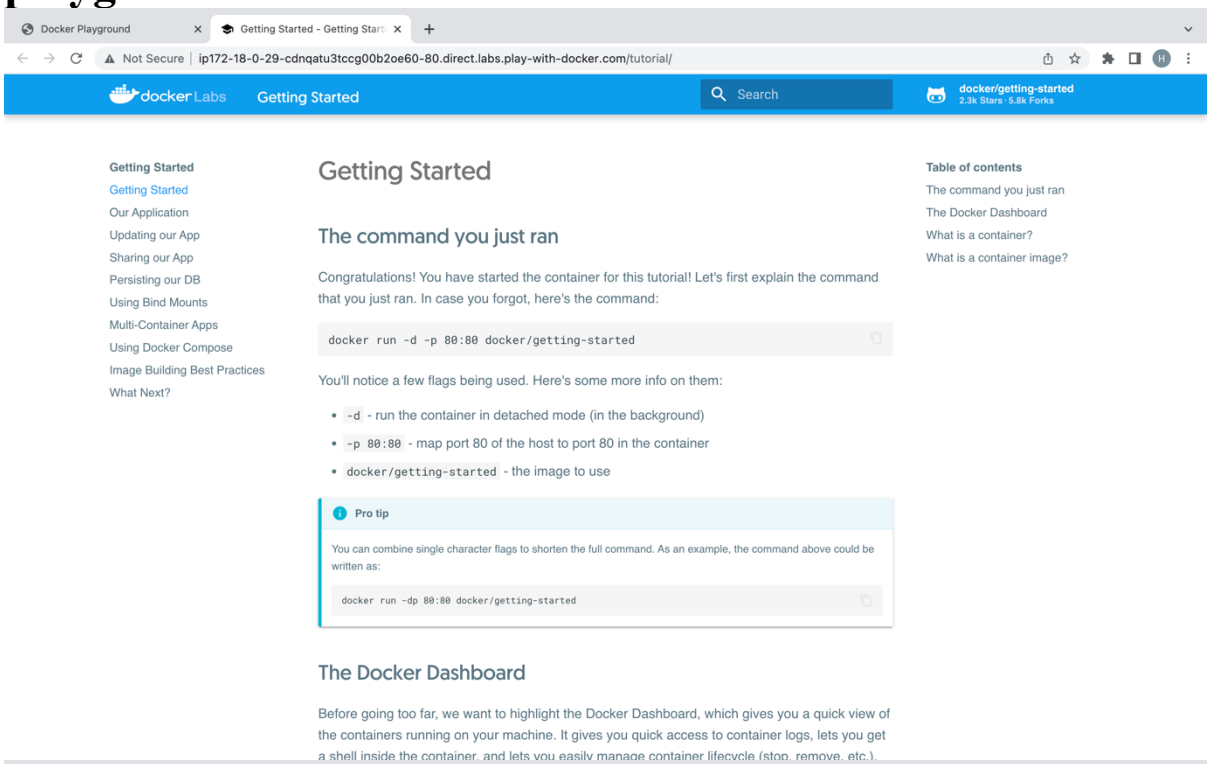
Question:

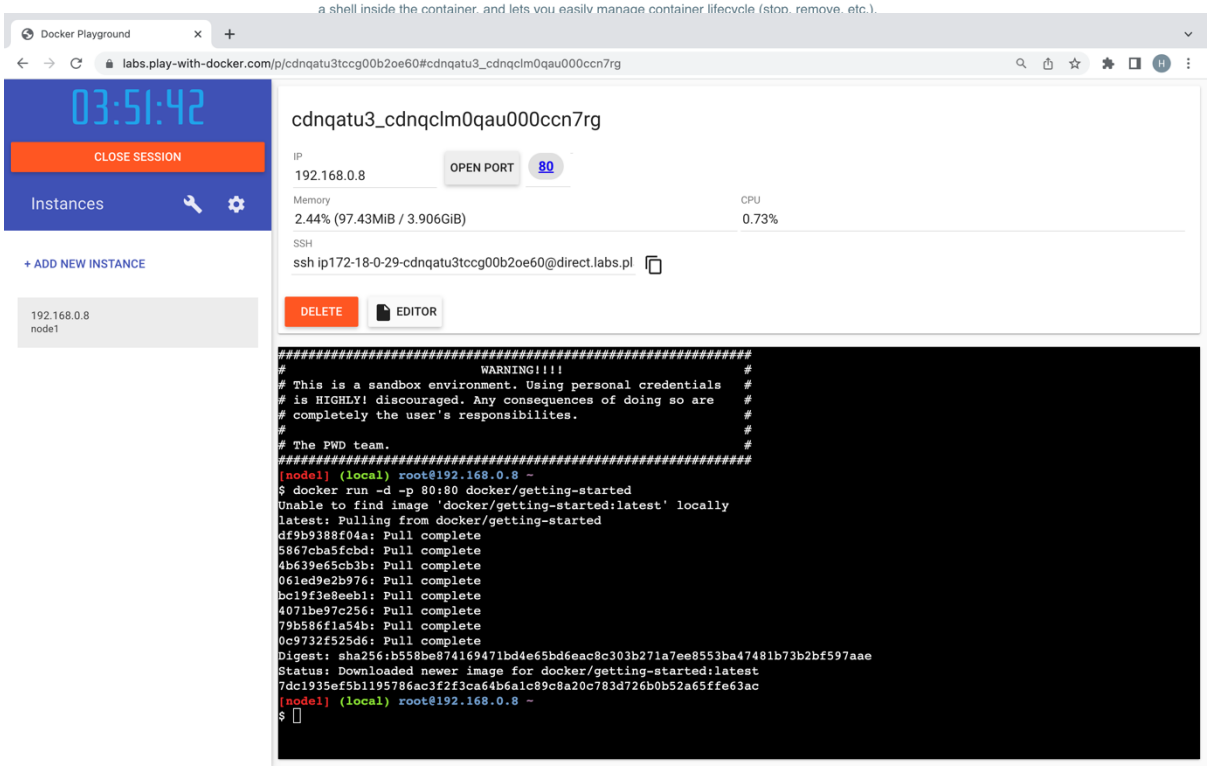
1.Pull an Image from docker hub and run it in docker playground.
2.Create a docker file for the jobportal application and deploy it in Docker desktop application.
3.Create a IBM container registry and deploy helloworld app or jobportalapp.
4.Create a Kubernetes cluster in IBM cloud and deploy helloworld image or jobportal image and also expose the same app to run in nodeport.

Solution:

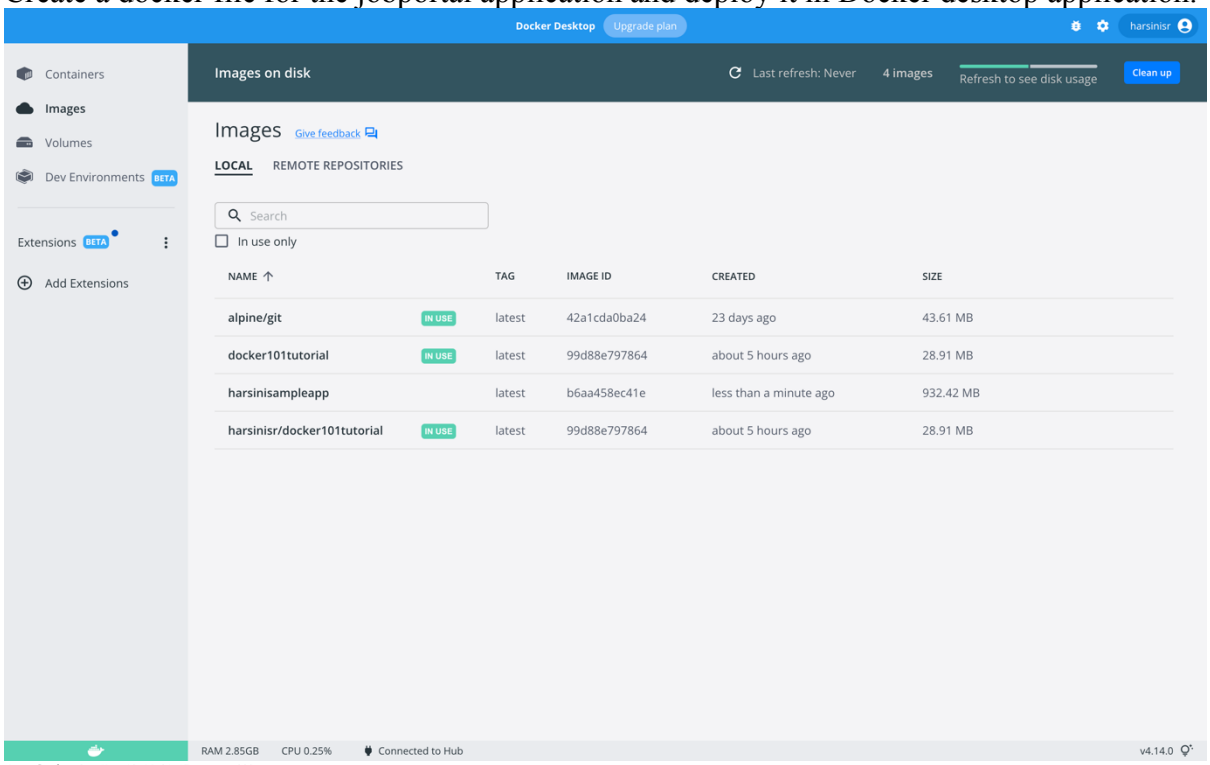


Pull an Image from docker hub and run it in docker playground.

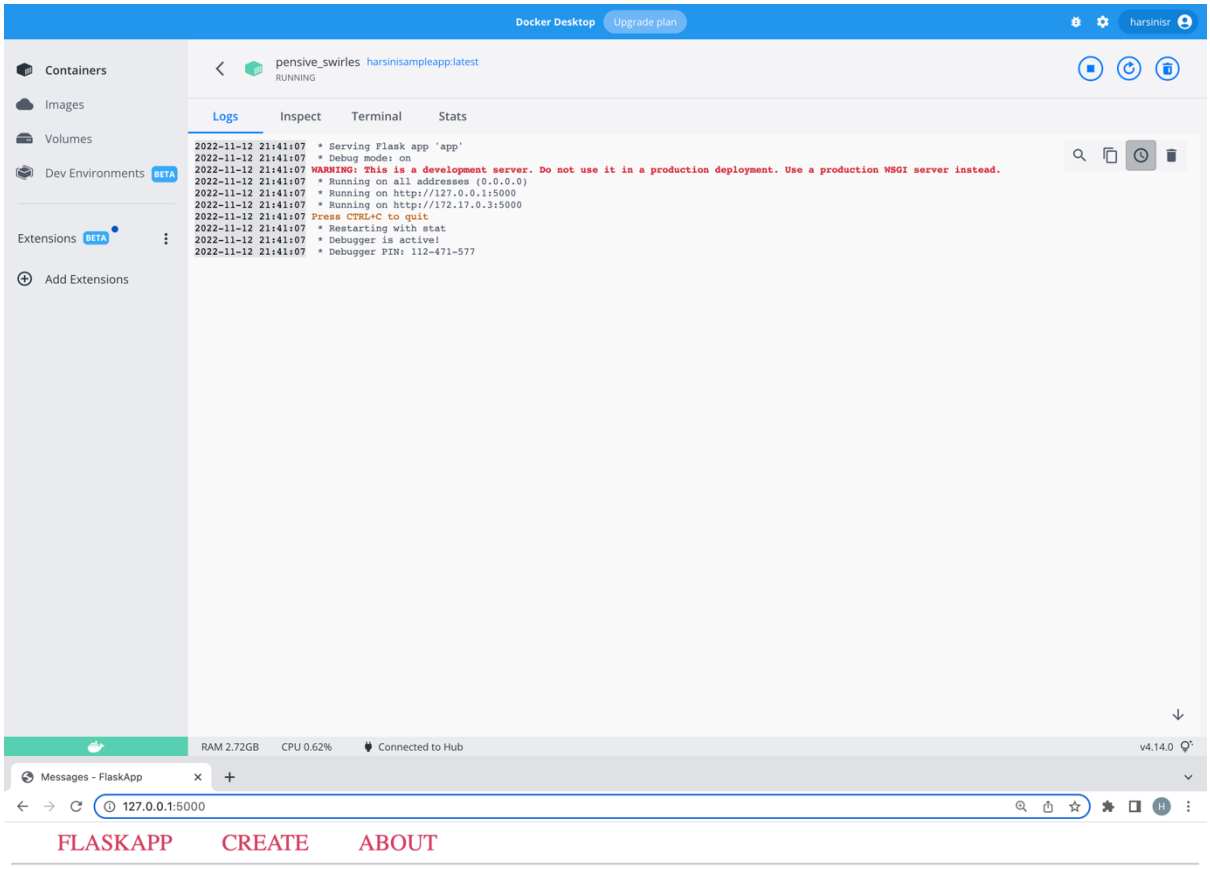




Create a docker file for the jobportal application and deploy it in Docker desktop application.

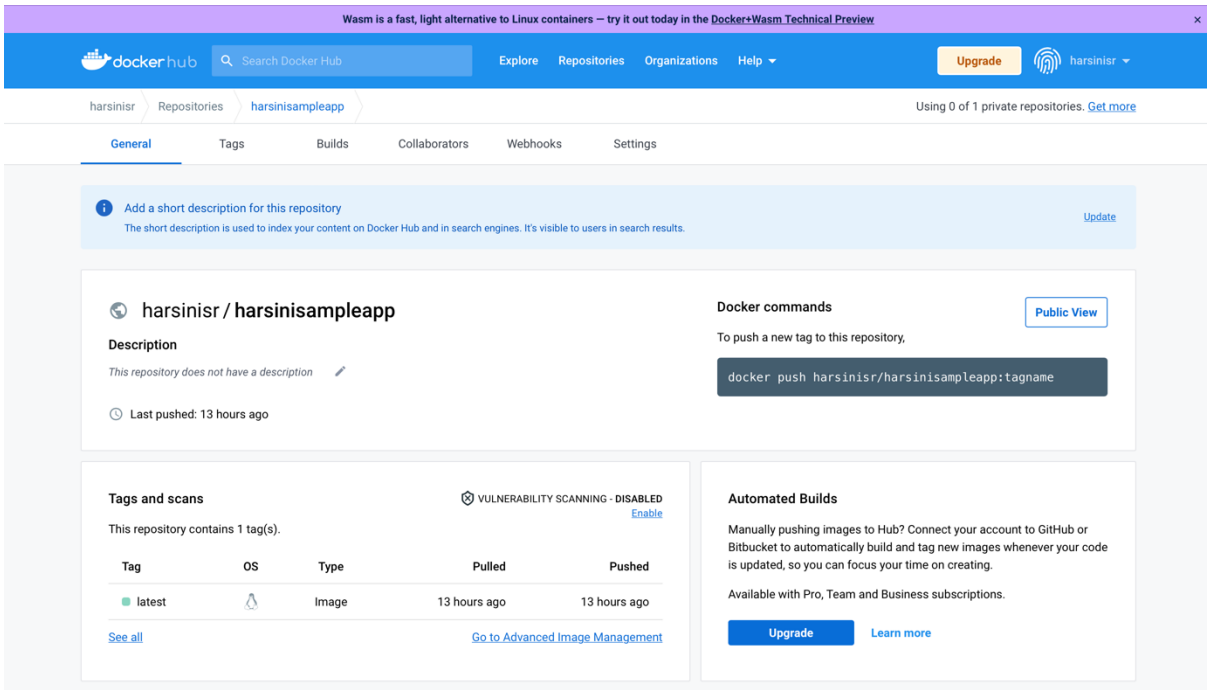


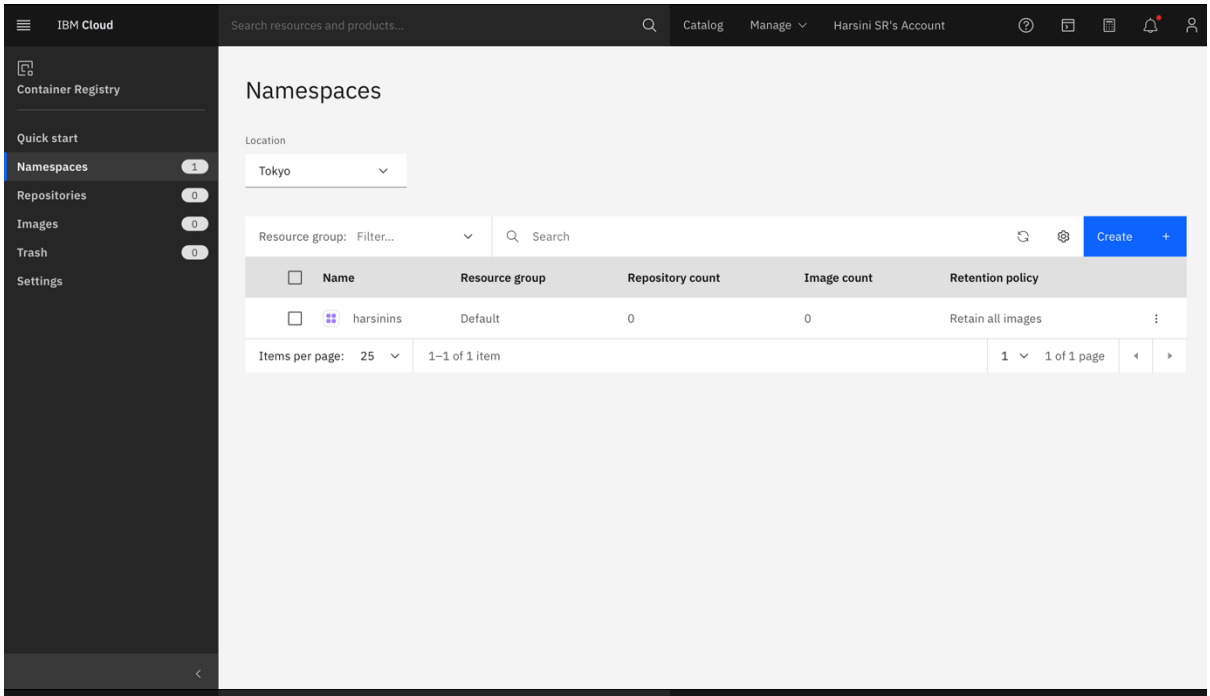

```
Last login: Sat Nov 12 16:31:17 on ttys000
harsini@Harsinis-MacBook-Pro ~ % cd desktop
harsini@Harsinis-MacBook-Pro desktop % cd flask_with_form_and_docker-main
harsini@Harsinis-MacBook-Pro flask_with_form_and_docker-main % docker build -t harsinisampleapp
"docker build" requires exactly 1 argument.
See 'docker build --help'.

Usage:  docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile
harsini@Harsinis-MacBook-Pro flask_with_form_and_docker-main % docker build -t harsinisampleapp .
[+] Building 55.7s (11/11) FINISHED
 => [internal] load build definition from Dockerfile                                                                                                0.1s
 => => transferring dockerfile: 267B                                                                                                                0.0s
 => [internal] load .dockerignore                                                                                                                   0.0s
 => => transferring context: 2B                                                                                                                     0.0s
 => [internal] load metadata for docker.io/library/python:3.10.6                                                                                    4.6s
 => [auth] library/python:pull token for registry-1.docker.io                                                                                       0.0s
 => [internal] load build context                                                                                                                   0.1s
 => => transferring context: 19.58kB                                                                                                                0.1s
 => [1/5] FROM docker.io/library/python:3.10.6@sha256:745efdfb7e4aac9a8422bd8c62d8bc35a693e8979a240d29677cb03e6aa91052                             44.1s
 => => resolve docker.io/library/python:3.10.6@sha256:745efdfb7e4aac9a8422bd8c62d8bc35a693e8979a240d29677cb03e6aa91052                              0.0s
 => => sha256:3e94d13e55e7a4ef17ff21376f57fb95c7e1706931f8704aa99260968d81f6e4 5.16MB / 5.16MB                                                      2.2s
 => => sha256:745efdfb7e4aac9a8422bd8c62d8bc35a693e8979a240d29677cb03e6aa91052 2.35kB / 2.35kB                                                      0.0s
 => => sha256:1671565cc8df8c365c9b661d3fbc164e73d01f1b0430c6179588428f99a9da2e 55.01MB / 55.01MB                                                    9.4s
 => => sha256:fa9c7528c685216129e8e67bf362a7702e7b1daa585ab85546a41508830657d6 10.88MB / 10.88MB                                                    3.8s
 => => sha256:8d1f943ceaaf3b3ce05df5c0926e7958836b048b700176bf9c56d8f37ac13fca 2.22kB / 2.22kB                                                      0.0s
 => => sha256:d25a66380b10283603ff696d7777bba5cdb1b9126fb0be7d118b9574946bcf84 8.53kB / 8.53kB                                                      0.0s
 => => sha256:53ad072f9cd16fc8eb93b182b20e758e11acc0ef60babef0bf1043c08de1901a 54.58MB / 54.58MB                                                    13.7s
 => => sha256:d6b983117533b718374f1701ef593dd2afa6613c7908c6553be8e2a150e6448a 196.79MB / 196.79MB                                                  35.0s
 => => sha256:d8092d56ded5476fe7c302256eb4dc6ff495ae8fb4dd28aa18dbcb7581e24a6c 6.29MB / 6.29MB                                                      11.4s
 => => extracting sha256:1671565cc8df8c365c9b661d3fbc164e73d01f1b0430c6179588428f99a9da2e                                                           3.6s
 => => sha256:c71afc637d59adc44c5fd3c348504df82b35bbb204f0057ea22c6ac8a1d285a5 20.02MB / 20.02MB                                                    15.8s
 => => extracting sha256:3e94d13e55e7a4ef17ff21376f57fb95c7e1706931f8704aa99260968d81f6e4                                                           0.4s
 => => sha256:864a10b3c704553e08cb5fcd12fbaee1c07048f6365f0fa35e84a285413da40b 234B / 234B                                                          14.0s
 => => extracting sha256:fa9c7528c685216129e8e67bf362a7702e7b1daa585ab85546a41508830657d6                                                           0.4s
 => => sha256:4334b2fe8293d19ddc1c3559093aae88f21601a7c85a31c6da6c0dc48fb6ed3c 3.04MB / 3.04MB                                                      15.3s
 => => extracting sha256:53ad072f9cd16fc8eb93b182b20e758e11acc0ef60babef0bf1043c08de1901a                                                           3.3s
 => => extracting sha256:d6b983117533b718374f1701ef593dd2afa6613c7908c6553be8e2a150e6448a                                                           7.1s
 => => extracting sha256:d8092d56ded5476fe7c302256eb4dc6ff495ae8fb4dd28aa18dbcb7581e24a6c                                                           0.3s
 => => extracting sha256:c71afc637d59adc44c5fd3c348504df82b35bbb204f0057ea22c6ac8a1d285a5                                                           0.6s
 => => extracting sha256:864a10b3c704553e08cb5fcd12fbaee1c07048f6365f0fa35e84a285413da40b                                                           0.0s
 => => extracting sha256:4334b2fe8293d19ddc1c3559093aae88f21601a7c85a31c6da6c0dc48fb6ed3c                                                           0.2s
 => [2/5] WORKDIR /app                                                                                                                              0.5s
 => [3/5] COPY requirements.txt ./                                                                                                                  0.0s
 => [4/5] RUN pip install -r requirements.txt                                                                                                       6.1s
 => [5/5] COPY . .                                                                                                                                  0.0s
 => exporting to image                                                                                                                              0.2s
 => => exporting layers                                                                                                                             0.2s
 => => writing image sha256:b6aa458ec41eb1f84735c1244d1c2f57cb4a3ea88ba62139ce0c2f3c75be36aa                                                        0.0s
 => => naming to docker.io/library/harsinisampleapp                                                                                                 0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
harsini@Harsinis-MacBook-Pro flask_with_form_and_docker-main %
```

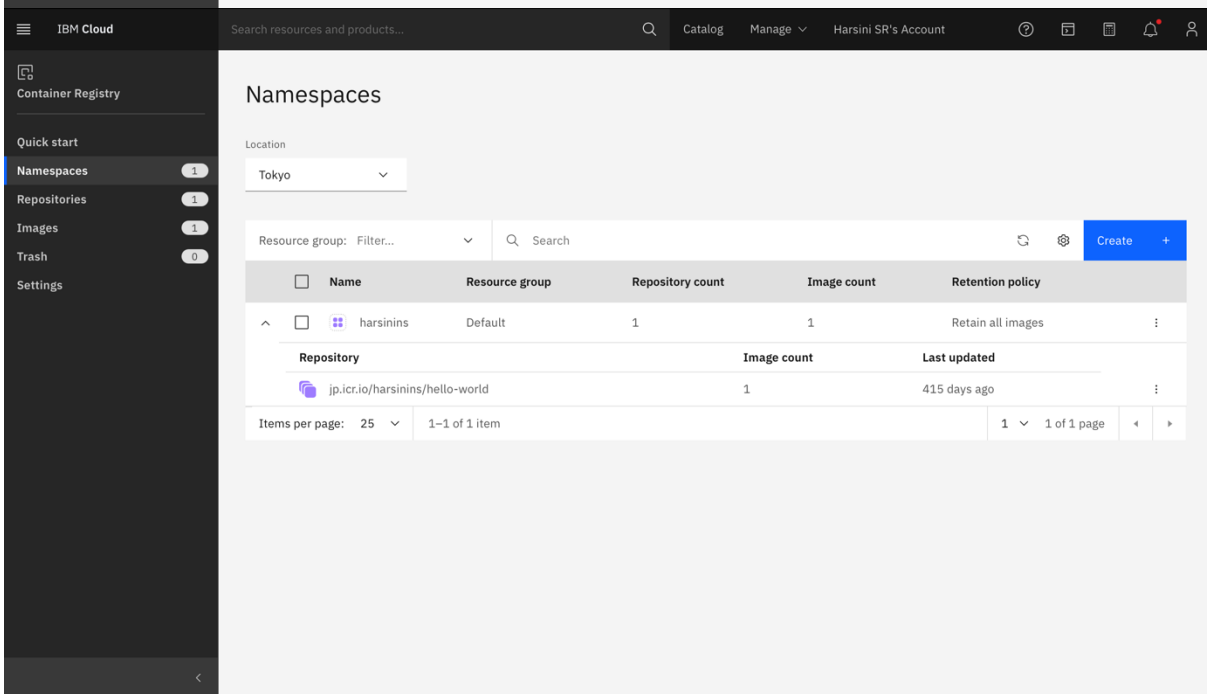


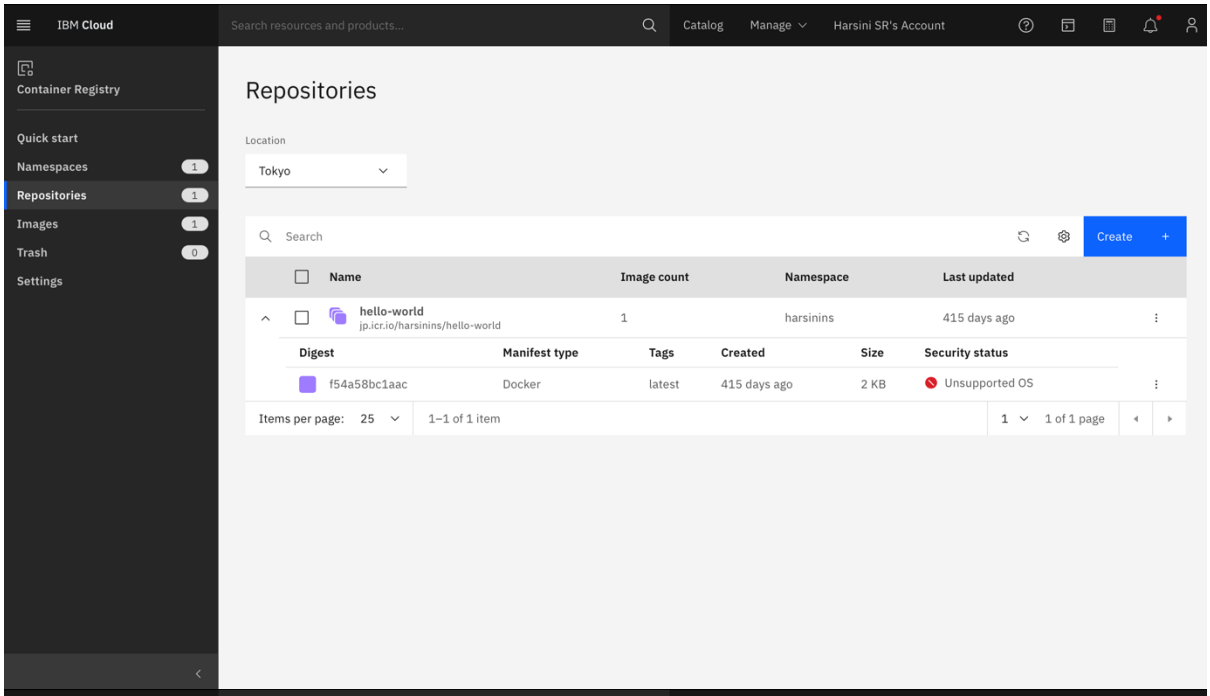
Create a IBM container registry and deploy helloworld app or jobportalapp

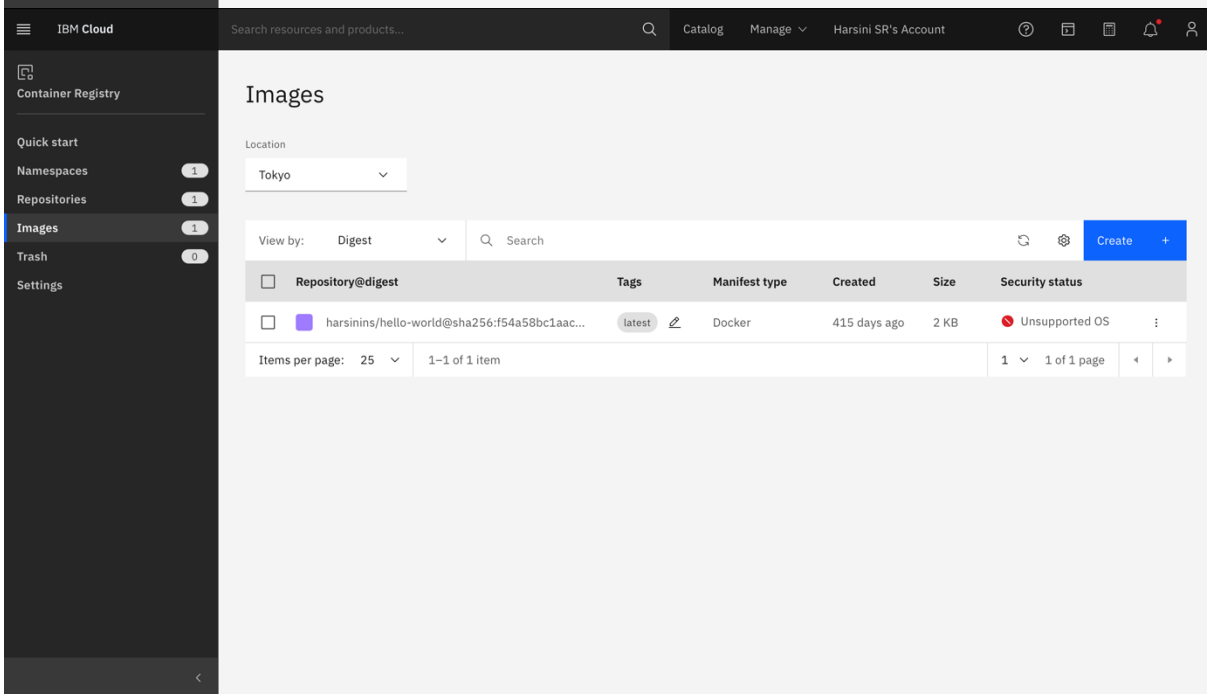
```
 => [3/5] COPY requirements.txt ./                                                                                    0.0s
 => [4/5] RUN pip install -r requirements.txt                                                                         6.1s
 => [5/5] COPY . .                                                                                                    0.0s
 => exporting to image                                                                                                0.2s
 => => exporting layers                                                                                               0.2s
 => => writing image sha256:b6aa458ec41eb1f84735c1244d1c2f57cb4a3ea88ba62139ce0c2f3c75be36aa                          0.0s
 => => naming to docker.io/library/harsinisampleapp                                                                   0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
harsini@Harsinis-MacBook-Pro flask_with_form_and_docker-main % docker build -t harsinisampleapp .
[+] Building 3.3s (11/11) FINISHED
 => [internal] load build definition from Dockerfile                                                                  0.0s
 => => transferring dockerfile: 120B                                                                                  0.0s
 => [internal] load .dockerignore                                                                                     0.0s
 => => transferring context: 2B                                                                                       0.0s
 => [internal] load metadata for docker.io/library/python:3.10.6                                                      3.1s
 => [auth] library/python:pull token for registry-1.docker.io                                                         0.0s
 => [1/5] FROM docker.io/library/python:3.10.6@sha256:745efdfb7e4aac9a8422bd8c62d8bc35a693e8979a240d29677cb03e6aa91052  0.0s
 => [internal] load build context                                                                                     0.0s
 => => transferring context: 9.72kB                                                                                   0.0s
 => CACHED [2/5] WORKDIR /app                                                                                         0.0s
 => CACHED [3/5] COPY requirements.txt ./                                                                             0.0s
 => CACHED [4/5] RUN pip install -r requirements.txt                                                                  0.0s
 => [5/5] COPY . .                                                                                                    0.0s
 => exporting to image                                                                                                0.0s
 => => exporting layers                                                                                               0.0s
 => => writing image sha256:a36bca38ed1db750c4e1ceeb884a3202b23916e3b0306f8fed9de51f7c93daf1                          0.0s
 => => naming to docker.io/library/harsinisampleapp                                                                   0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
harsini@Harsinis-MacBook-Pro flask_with_form_and_docker-main % docker run -p 5000:5000 harsinisampleapp
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.17.0.2:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 874-697-760
172.17.0.1 - - [12/Nov/2022 16:21:15] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [12/Nov/2022 16:21:16] "GET /favicon.ico HTTP/1.1" 404 -
^C
harsini@Harsinis-MacBook-Pro flask_with_form_and_docker-main % docker tag harsinisampleapp harsinisr/harsinisampleapp
harsini@Harsinis-MacBook-Pro flask_with_form_and_docker-main % docker push harsinisr/harsinisampleapp
Using default tag: latest
The push refers to repository [docker.io/harsinisr/harsinisampleapp]
653b2d0bfd19: Pushed
0ce4d5f66c8a: Pushed
424f11dae8da: Pushed
21fb379aef9e: Pushed
bfc1deb8136e: Mounted from library/python
1f123186824c: Mounted from library/python
3d6eb1152931: Mounted from library/python
100796cdf3b1: Mounted from library/python
54acb5a6fa0b: Mounted from library/python
8d51c618126f: Mounted from library/python
9ff6e4d46744: Mounted from library/python
a89d1d47b5a1: Mounted from library/python
655ed1b7a428: Mounted from library/python
latest: digest: sha256:395dc476a0174d7d8c59fd8b4cab61613bd402aa17419bbdb374808f2de71a0c size: 3050
harsini@Harsinis-MacBook-Pro flask_with_form_and_docker-main %
```




IBM Cloud
Search resources and products...
Catalog
Manage
Harsini SR's Account
Container Registry
Quick start
Namespaces 1
Repositories 0
Images 0
Trash 0
Settings
Namespaces
Location
Tokyo
Resource group: Filter...
Search
Create
Name
Resource group
Repository count
Image count
Retention policy
harsinins
Default
0
0
Retain all images
Items per page: 25
1–1 of 1 item
1 of 1 page

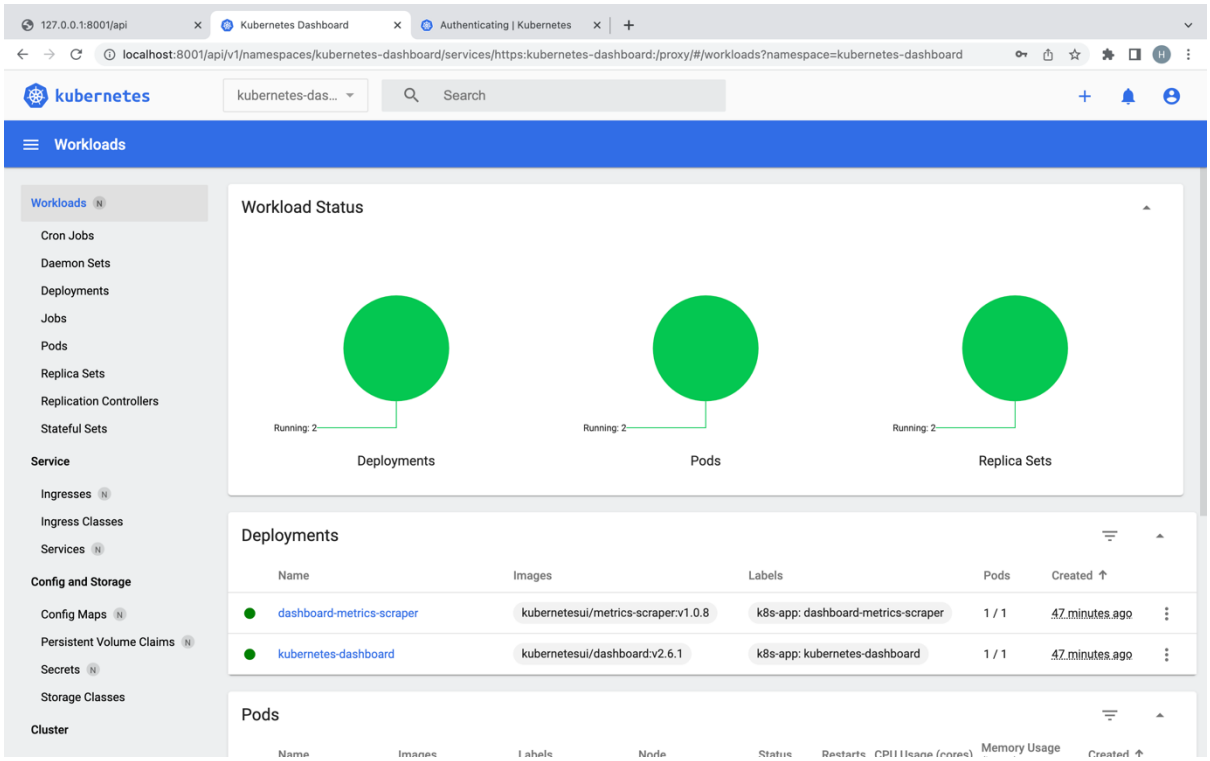

IBM Cloud
Search resources and products...
Catalog
Manage
Harsini SR's Account
Container Registry
Quick start
Namespaces 1
Repositories 1
Images 1
Trash 0
Settings
Namespaces
Location
Tokyo
Resource group: Filter...
Search
Create
Name
Resource group
Repository count
Image count
Retention policy
harsinins
Default
1
1
Retain all images
Repository
Image count
Last updated
jp.icr.io/harsinins/hello-world
1
415 days ago
Items per page: 25
1–1 of 1 item
1 of 1 page


IBM Cloud
Search resources and products...
Catalog
Manage
Harsini SR's Account
Container Registry
Quick start
Namespaces 1
Repositories 1
Images 1
Trash 0
Settings
Repositories
Location
Tokyo
Search
Create
Name
Image count
Namespace
Last updated
hello-world
jp.icr.io/harsinins/hello-world
1
harsinins
415 days ago
Digest
Manifest type
Tags
Created
Size
Security status
f54a58bc1aac
Docker
latest
415 days ago
2 KB
Unsupported OS
Items per page: 25
1–1 of 1 item
1
1 of 1 page


IBM Cloud
Search resources and products...
Catalog
Manage
Harsini SR's Account
Container Registry
Quick start
Namespaces 1
Repositories 1
Images 1
Trash 0
Settings
Images
Location
Tokyo
View by:
Digest
Search
Create
Repository@digest
Tags
Manifest type
Created
Size
Security status
harsinins/hello-world@sha256:f54a58bc1aac...
latest
Docker
415 days ago
2 KB
Unsupported OS
Items per page: 25
1–1 of 1 item
1
1 of 1 page

Create a Kubernetes cluster in IBM cloud and deploy helloworld image

```
  drain           Drain node in preparation for maintenance
  taint           Update the taints on one or more nodes

Troubleshooting and Debugging Commands:
  describe        Show details of a specific resource or group of resources
  logs            Print the logs for a container in a pod
  attach          Attach to a running container
  exec            Execute a command in a container
  port-forward    Forward one or more local ports to a pod
  proxy           Run a proxy to the Kubernetes API server
  cp              Copy files and directories to and from containers
  auth            Inspect authorization
  debug           Create debugging sessions for troubleshooting workloads and
nodes

Advanced Commands:
  diff            Diff the live version against a would-be applied version
  apply           Apply a configuration to a resource by file name or stdin
  patch           Update fields of a resource
  replace         Replace a resource by file name or stdin
  wait            Experimental: Wait for a specific condition on one or many
resources
  kustomize       Build a kustomization target from a directory or URL.

Settings Commands:
  label           Update the labels on a resource
  annotate        Update the annotations on a resource
  completion      Output shell completion code for the specified shell (bash,
zsh, fish, or powershell)

Other Commands:
  alpha           Commands for features in alpha
  api-resources   Print the supported API resources on the server
  api-versions    Print the supported API versions on the server, in the form of
"group/version"
  config          Modify kubeconfig files
  plugin          Provides utilities for interacting with plugins
  version         Print the client and server version information

Usage:
  kubectl [flags] [options]

Use "kubectl <command> --help" for more information about a given command.
Use "kubectl options" for a list of global command-line options (applies to all
commands).
harsini@Harsinis-MacBook-Pro ~ % kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.6.1/aio/deploy/recommended.yaml
namespace/kubernetes-dashboard created
serviceaccount/kubernetes-dashboard created
service/kubernetes-dashboard created
secret/kubernetes-dashboard-certs created
secret/kubernetes-dashboard-csrf created
secret/kubernetes-dashboard-key-holder created
configmap/kubernetes-dashboard-settings created
role.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
deployment.apps/kubernetes-dashboard created
service/dashboard-metrics-scraper created
deployment.apps/dashboard-metrics-scraper created
harsini@Harsinis-MacBook-Pro ~ % kubectl proxy
Starting to serve on 127.0.0.1:8001
```



```
Other Commands:
  alpha          Commands for features in alpha
  api-resources  Print the supported API resources on the server
  api-versions   Print the supported API versions on the server, in the form of "group/version"
  config         Modify kubeconfig files
  plugin         Provides utilities for interacting with plugins
  version        Print the client and server version information

Usage:
  kubectl [flags] [options]

Use "kubectl <command> --help" for more information about a given command.
Use "kubectl options" for a list of global command-line options (applies to all commands).
harsini@Harsinis-MacBook-Pro desktop % kubectl get nodes
NAME             STATUS   ROLES           AGE   VERSION
docker-desktop   Ready    control-plane   49m   v1.25.2
harsini@Harsinis-MacBook-Pro desktop % kubectl get pods
No resources found in default namespace.
harsini@Harsinis-MacBook-Pro desktop % kubectl get ns
NAME                   STATUS   AGE
default                Active   49m
kube-node-lease        Active   49m
kube-public            Active   49m
kube-system            Active   49m
kubernetes-dashboard   Active   39m
harsini@Harsinis-MacBook-Pro desktop % kubectl -n kubernetes-dashboard describe serviceaccount admin-user
Error from server (NotFound): serviceaccounts "admin-user" not found
harsini@Harsinis-MacBook-Pro desktop % kubectl apply -f dashboard-adminuser.yaml
error: the path "dashboard-adminuser.yaml" does not exist
harsini@Harsinis-MacBook-Pro desktop % cd flask_with_form_and_docker-main
harsini@Harsinis-MacBook-Pro flask_with_form_and_docker-main % cd kubernetes
harsini@Harsinis-MacBook-Pro kubernetes % kubectl apply -f dashboard-adminuser.yaml
serviceaccount/admin-user created
secret/admin-user-token created
clusterrolebinding.rbac.authorization.k8s.io/admin-user created
harsini@Harsinis-MacBook-Pro kubernetes % kubectl -n kubernetes-dashboard describe serviceaccount admin-user
Name:                admin-user
Namespace:           kubernetes-dashboard
Labels:              <none>
Annotations:         <none>
Image pull secrets:  <none>
Mountable secrets:   <none>
Tokens:              admin-user-token
Events:              <none>
harsini@Harsinis-MacBook-Pro kubernetes % kubectl -n kubernetes-dashboard describe secret admin-user-token
Name:         admin-user-token
Namespace:    kubernetes-dashboard
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: admin-user
              kubernetes.io/service-account.uid: e744c9ac-adea-48b8-ad2a-e6e2ebc7d952

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1099 bytes
namespace:  20 bytes
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6IkRwTjVOMXdWYVFBQW16ZVlfVkVtLW5adDZvWFRUU2lVdzdIcWtJUGpxOXcifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJrdWJlcm5ldGVzLWRhc2hib2FyZCIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJhZG1pbi11c2VyLXRva2VuIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQubmFtZSI6ImFkbWluLXVzZXIiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC51aWQiOiJlNzQ0YzlhYy1hZGVhLTQ4YjgtYWQyYS1lNmUyZWJjN2Q5NTIiLCJzdWIiOiJzeXN0ZW06c2VydmljZWFjY291bnQ6a3ViZXJuZXRlcy1kYXNoYm9hcmQ6YWRtaW4tdXNlciJ9.Q4SX
```

FLASKAPP CREATE ABOUT

Blog Page

Messages

Message One

Message One Content

Message Two

Message Two Content

dashboard-adminuser.yaml

```yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kubernetes-dashboard
---
apiVersion: v1
kind: Secret
metadata:
```

```
  name: admin-user-token
  namespace: kubernetes-dashboard
  annotations:
    kubernetes.io/service-account.name: admin-user
type: kubernetes.io/service-account-token

---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard
```

flask_deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask-app

spec:
  replicas: 1
  selector:
    matchLabels:
      app: flask-app
  template:
    metadata:
      labels:
        app: flask-app

    spec:
      containers:
        - name: flask-app-container
          image: flask-app
          imagePullPolicy: Never
          ports:
            - containerPort: 5000
              protocol: TCP
```

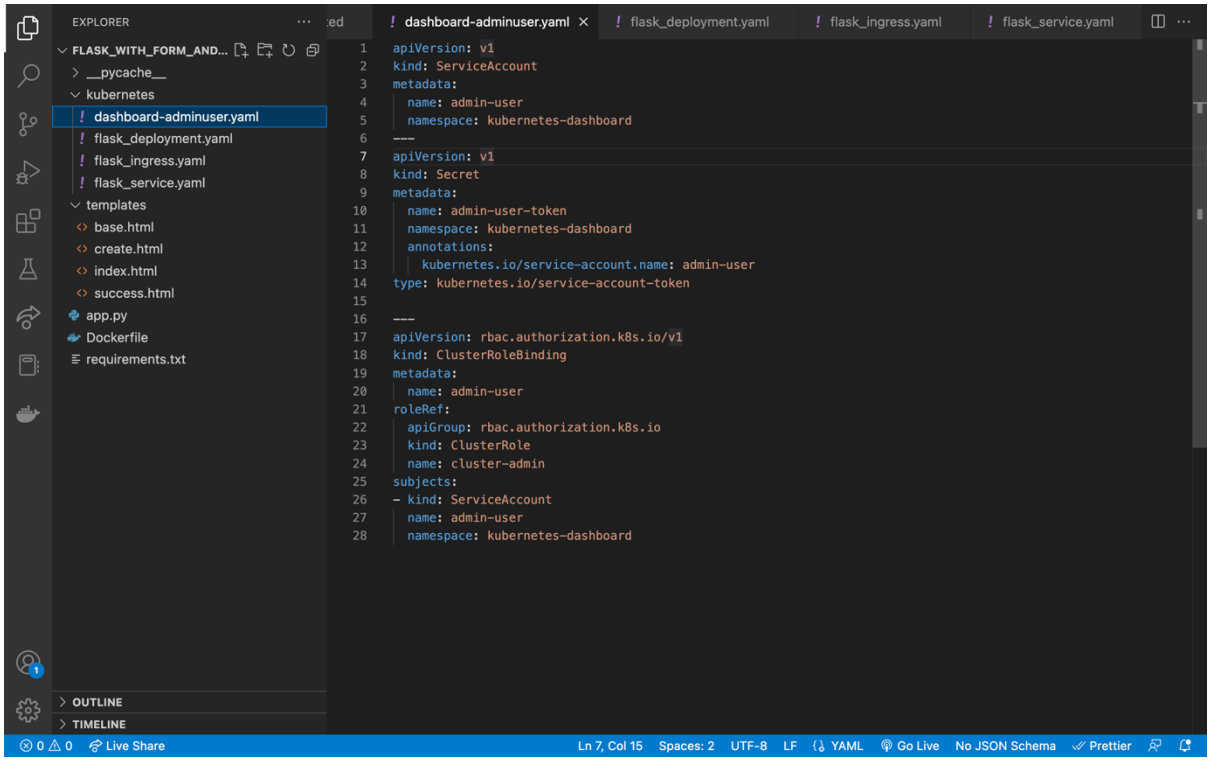
flask_ingress.yaml

```
apiVersion: networking.k8s.io/v1
kind: Ingress
```

```yaml
metadata:
  name: flask-app-ingress
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/ssl-redirect: "false"

spec:
  rules:
    - http:
        paths:
          - backend:
              service:
                name: flask-app-service
                port:
                  number: 5000
            path: /
            pathType: Prefix
```

flask_service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: flask-app-service
spec:
  type: ClusterIP
  ports:
    - port: 5000
  selector:
    app: flask-app
```


EXPLORER
FLASK_WITH_FORM_AND...
__pycache__
kubernetes
dashboard-adminuser.yaml
flask_deployment.yaml
flask_ingress.yaml
flask_service.yaml
templates
base.html
create.html
index.html
success.html
app.py
Dockerfile
requirements.txt
OUTLINE
TIMELINE
dashboard-adminuser.yaml
flask_deployment.yaml
flask_ingress.yaml
flask_service.yaml
apiVersion: v1
kind: ServiceAccount
metadata:
name: admin-user
namespace: kubernetes-dashboard
---
apiVersion: v1
kind: Secret
metadata:
name: admin-user-token
namespace: kubernetes-dashboard
annotations:
kubernetes.io/service-account.name: admin-user
type: kubernetes.io/service-account-token
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
name: admin-user
roleRef:
apiGroup: rbac.authorization.k8s.io
kind: ClusterRole
name: cluster-admin
subjects:
- kind: ServiceAccount
name: admin-user
namespace: kubernetes-dashboard
0 0
Live Share
Ln 7, Col 15
Spaces: 2
UTF-8
LF
YAML
Go Live
No JSON Schema
Prettier