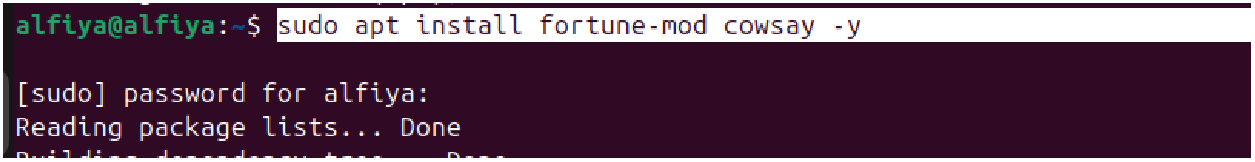
Problem Statement 1

1. Downloading the Prerequisites

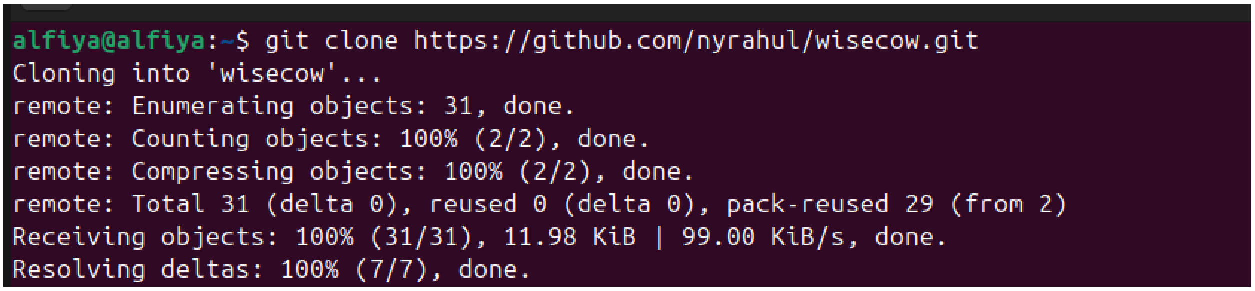
```
sudo apt install fortune-mod cowsay -y
```



```
alfiya@alfiya:~$ sudo apt install fortune-mod cowsay -y

[sudo] password for alfiya:
Reading package lists... Done
Building dependency tree... Done
```

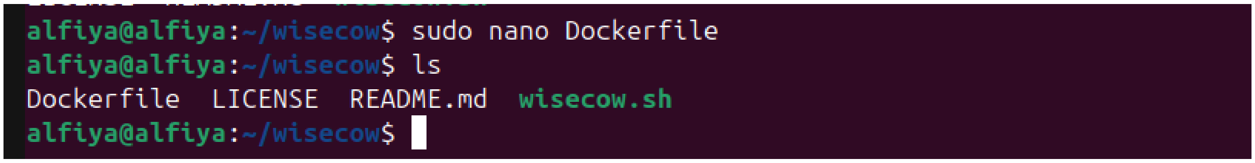
2. Clone the Wisecow repo

```
git clone https://github.com/nyrahul/wisecow.git
cd wisecow
```

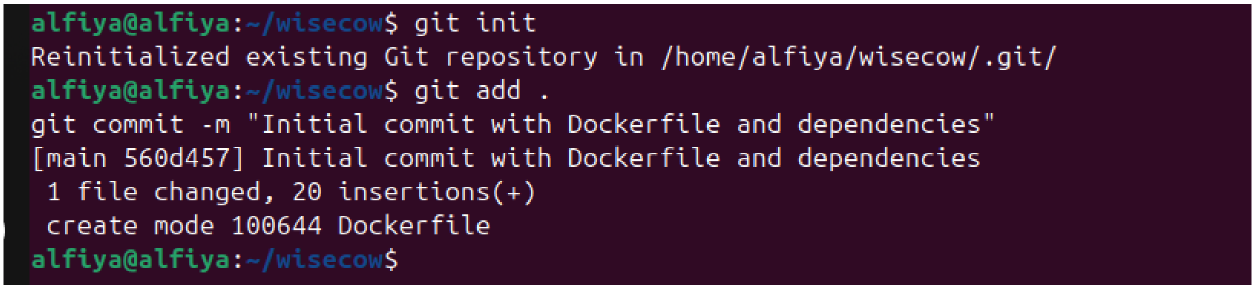


```
alfiya@alfiya:~$ git clone https://github.com/nyrahul/wisecow.git
Cloning into 'wisecow'...
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 31 (delta 0), reused 0 (delta 0), pack-reused 29 (from 2)
Receiving objects: 100% (31/31), 11.98 KiB | 99.00 KiB/s, done.
Resolving deltas: 100% (7/7), done.
```

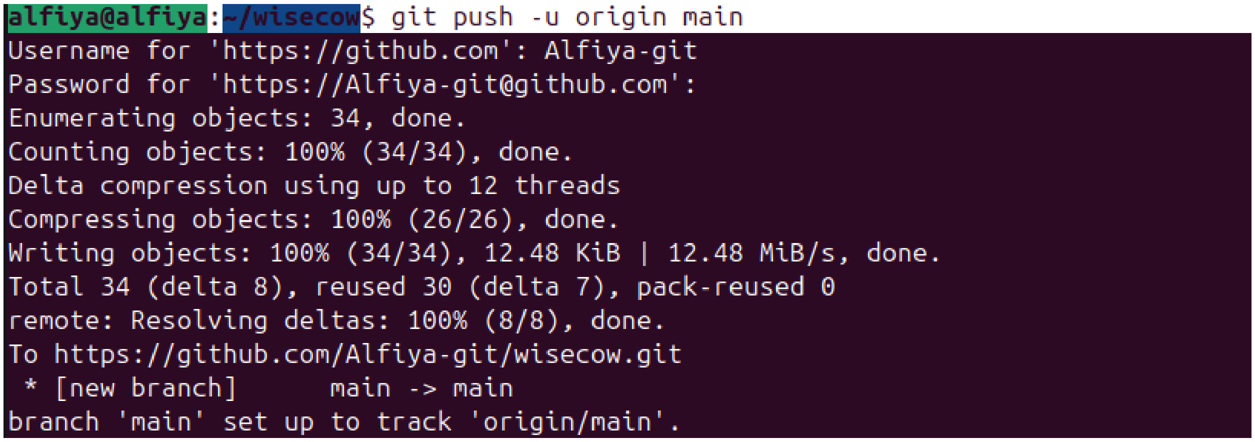
3. Create a Dockerfile, set up a Git repository, and push the Dockerfile along with supporting files to the repository.
   NOTE: Even though dependencies were already installed locally, I included them in the Dockerfile to ensure a reproducible and portable environment, which is considered a best practice.



```
alfiya@alfiya:~/wisecow$ sudo nano Dockerfile
alfiya@alfiya:~/wisecow$ ls
Dockerfile  LICENSE  README.md  wisecow.sh
alfiya@alfiya:~/wisecow$ 
```



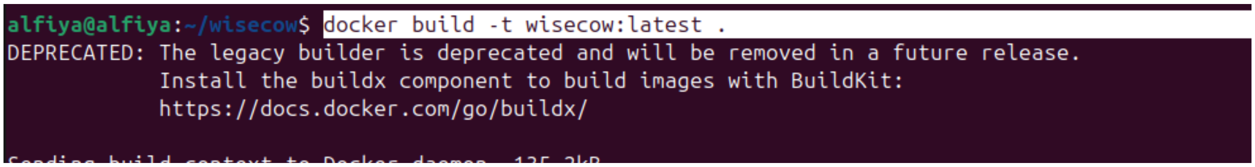
```
alfiya@alfiya:~/wisecow$ git init
Reinitialized existing Git repository in /home/alfiya/wisecow/.git/
alfiya@alfiya:~/wisecow$ git add .
git commit -m "Initial commit with Dockerfile and dependencies"
[main 560d457] Initial commit with Dockerfile and dependencies
 1 file changed, 20 insertions(+)
 create mode 100644 Dockerfile
alfiya@alfiya:~/wisecow$
```



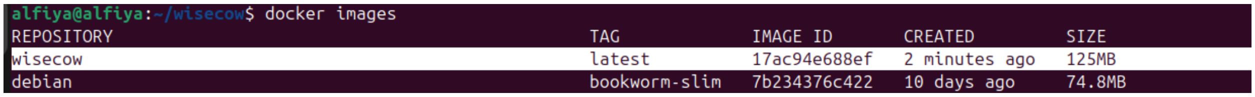
```
alfiya@alfiya:~/wisecow$ git push -u origin main
Username for 'https://github.com': Alfiya-git
Password for 'https://Alfiya-git@github.com':
Enumerating objects: 34, done.
Counting objects: 100% (34/34), done.
Delta compression using up to 12 threads
Compressing objects: 100% (26/26), done.
Writing objects: 100% (34/34), 12.48 KiB | 12.48 MiB/s, done.
Total 34 (delta 8), reused 30 (delta 7), pack-reused 0
remote: Resolving deltas: 100% (8/8), done.
To https://github.com/Alfiya-git/wisecow.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

4. Build and test locally

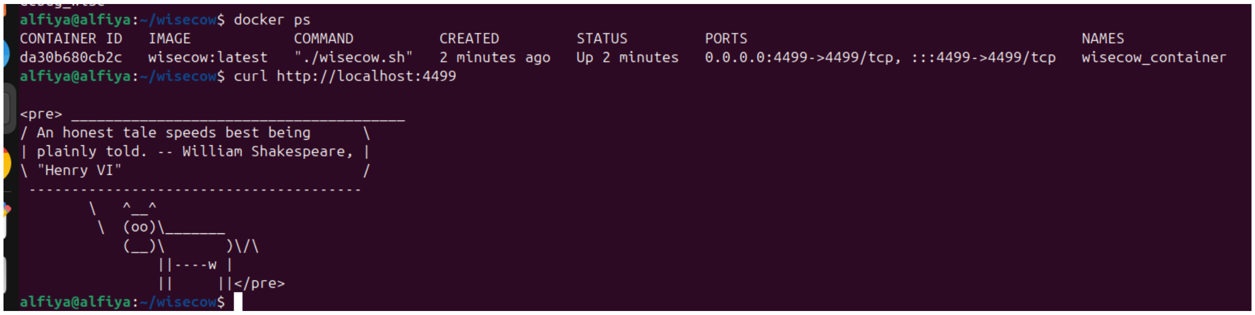
```
docker build -t wisecow:latest .
docker run -p 4499:4499 --name wisecow-container wisecow:latest
```



```
alfiya@alfiya:~/wisecow$ docker build -t wisecow:latest .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  125.2kB
```



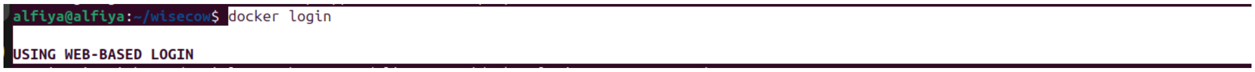
```
alfiya@alfiya:~/wisecow$ docker images
REPOSITORY                                                    TAG             IMAGE ID       CREATED         SIZE
wisecow                                                       latest          17ac94e688ef   2 minutes ago   125MB
debian                                                        bookworm-slim   7b234376c422   10 days ago     74.8MB
```



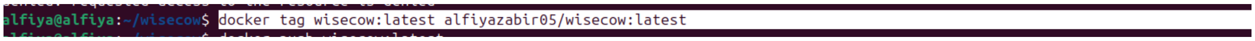
```
alfiya@alfiya:~/wisecow$ docker ps
CONTAINER ID   IMAGE            COMMAND            CREATED         STATUS         PORTS                                       NAMES
da30b680cb2c   wisecow:latest   "./wisecow.sh"     2 minutes ago   Up 2 minutes   0.0.0.0:4499->4499/tcp, :::4499->4499/tcp   wisecow_container
alfiya@alfiya:~/wisecow$ curl http://localhost:4499

<pre> _______________________________________
/ An honest tale speeds best being      \
| plainly told. -- William Shakespeare, |
\ "Henry VI"                            /
 ---------------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||</pre>
alfiya@alfiya:~/wisecow$ 
```

5. Pushing the image to Docker Hub.
    - Enter your Docker Hub username and password
    - Tag your local image
    - Push to Docker hub

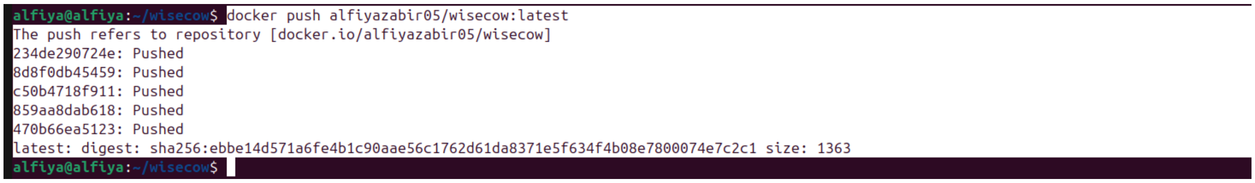
```
docker login
docker tag image:tag username/image:tag
docker push username/image:tag
```



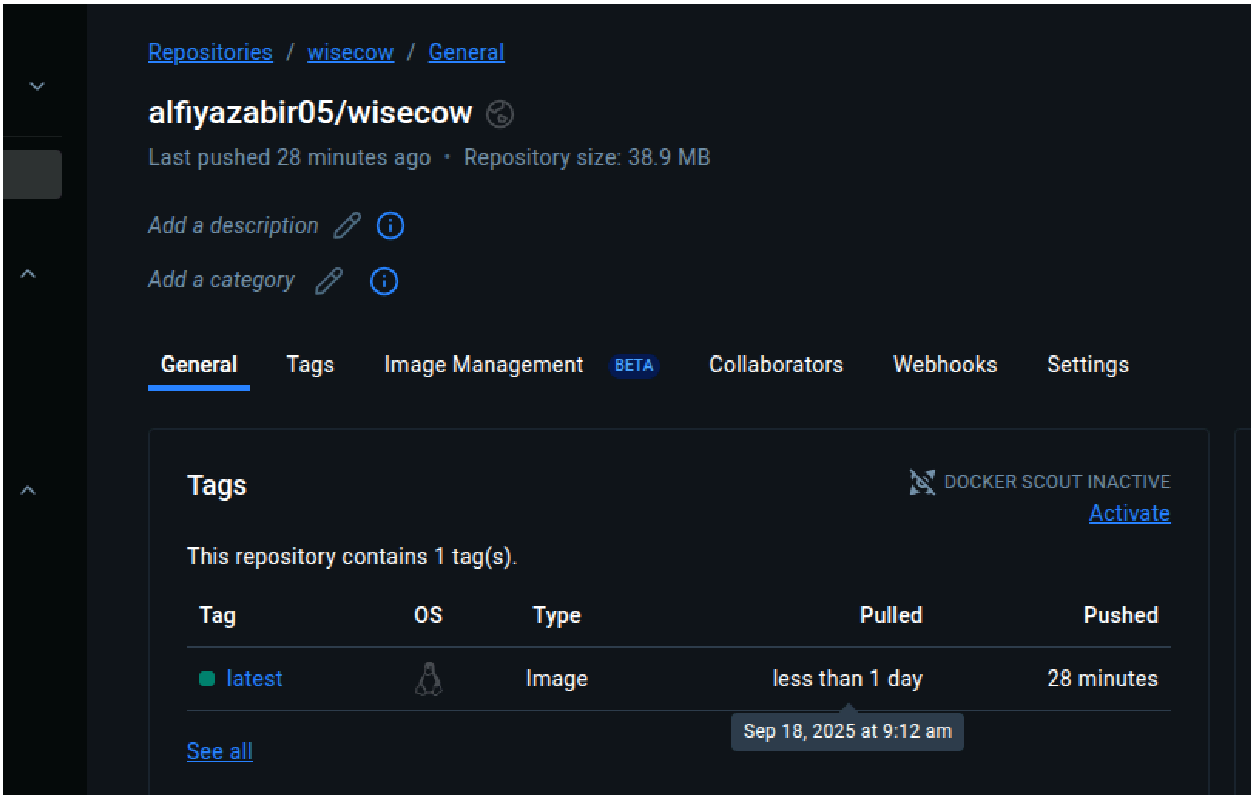
```
alfiya@alfiya:~/wisecow$ docker login

USING WEB-BASED LOGIN
```



```
alfiya@alfiya:~/wisecow$ docker tag wisecow:latest alfiyazabir05/wisecow:latest
```



```
alfiya@alfiya:~/wisecow$ docker push alfiyazabir05/wisecow:latest
The push refers to repository [docker.io/alfiyazabir05/wisecow]
234de290724e: Pushed
8d8f0db45459: Pushed
c50b4718f911: Pushed
859aa8dab618: Pushed
470b66ea5123: Pushed
latest: digest: sha256:ebbe14d571a6fe4b1c90aae56c1762d61da8371e5f634f4b08e7800074e7c2c1 size: 1363
alfiya@alfiya:~/wisecow$
```



Repositories / wisecow / General

alfiyazabir05/wisecow

Last pushed 28 minutes ago • Repository size: 38.9 MB

Add a description

Add a category

Tags

DOCKER SCOUT INACTIVE Activate

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	less than 1 day	28 minutes

Sep 18, 2025 at 9:12 am

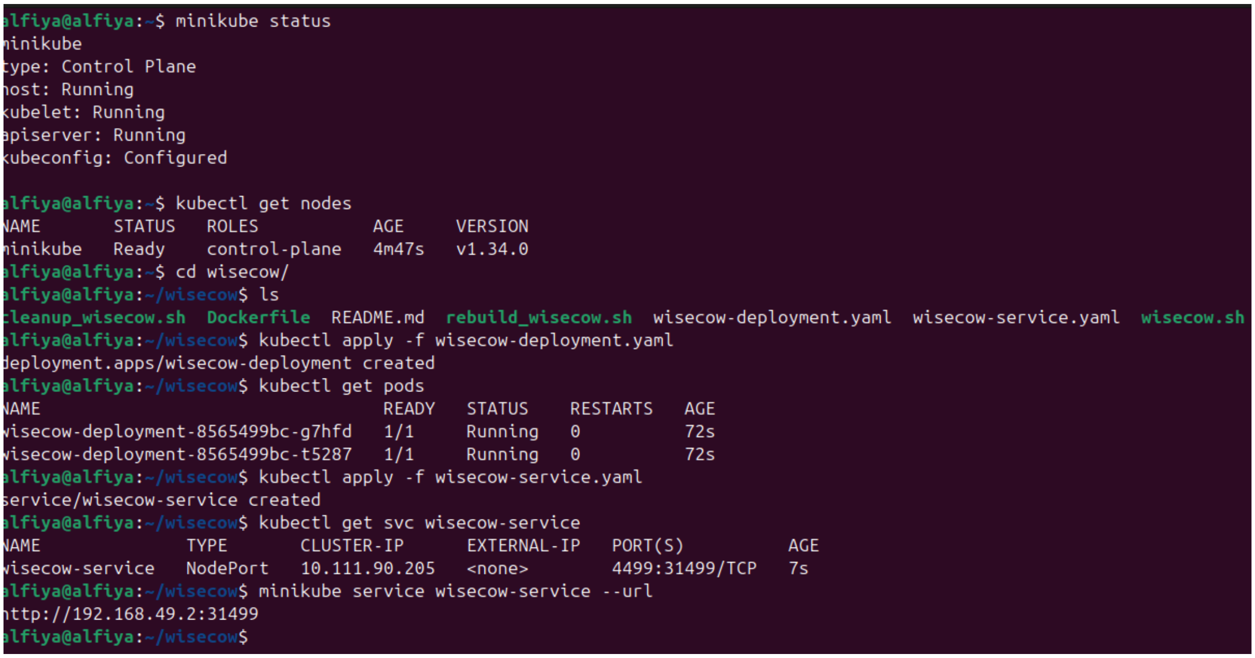
See all

6. Create Kubernetes Deployment Manifest (wisecow-deployment.yaml) and Kubernetes Service Manifest (wisecow-service.yaml)
7. Deploy on kubernetes (since i have already installed minikube, kubectl)

```
kubectl apply -f wisecow-deployment.yaml
kubectl apply -f wisecow-service.yaml
```

8. Check that pods and service are running

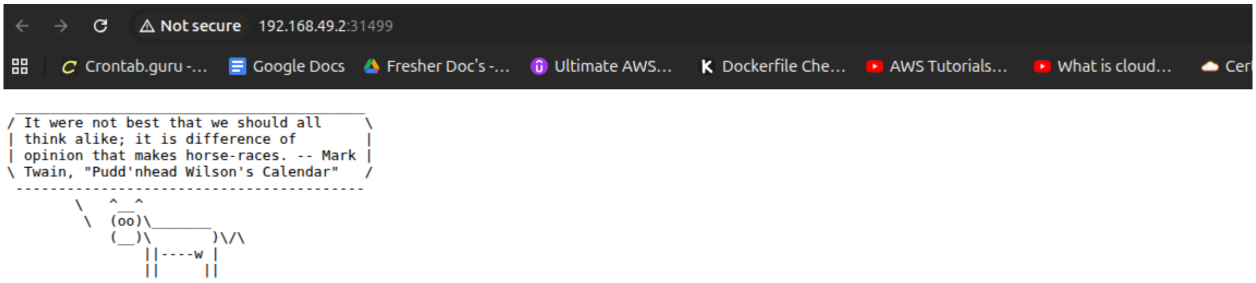
```
kubectl get pods
kubectl get svc wisecow-service
```



```
alfiya@alfiya:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

alfiya@alfiya:~$ kubectl get nodes
NAME       STATUS   ROLES           AGE     VERSION
minikube   Ready    control-plane   4m47s   v1.34.0
alfiya@alfiya:~$ cd wisecow/
alfiya@alfiya:~/wisecow$ ls
cleanup_wisecow.sh  Dockerfile  README.md  rebuild_wisecow.sh  wisecow-deployment.yaml  wisecow-service.yaml  wisecow.sh
alfiya@alfiya:~/wisecow$ kubectl apply -f wisecow-deployment.yaml
deployment.apps/wisecow-deployment created
alfiya@alfiya:~/wisecow$ kubectl get pods
NAME                                  READY   STATUS    RESTARTS   AGE
wisecow-deployment-8565499bc-g7hfd    1/1     Running   0          72s
wisecow-deployment-8565499bc-t5287    1/1     Running   0          72s
alfiya@alfiya:~/wisecow$ kubectl apply -f wisecow-service.yaml
service/wisecow-service created
alfiya@alfiya:~/wisecow$ kubectl get svc wisecow-service
NAME              TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
wisecow-service   NodePort   10.111.90.205   <none>        4499:31499/TCP   7s
alfiya@alfiya:~/wisecow$ minikube service wisecow-service --url
http://192.168.49.2:31499
alfiya@alfiya:~/wisecow$
```

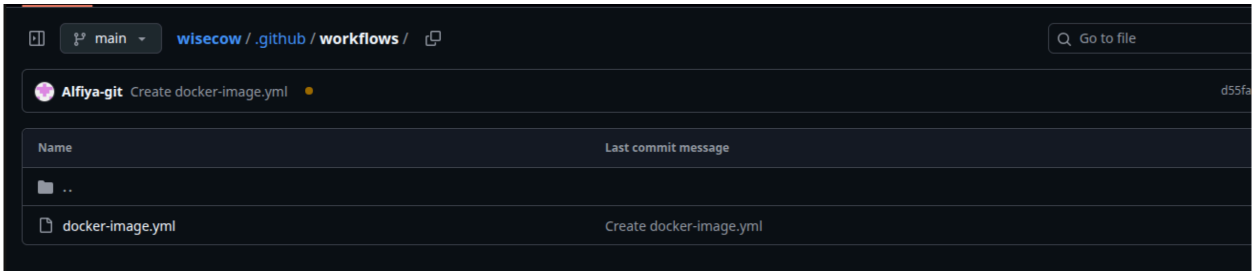
9. Get the service URL (to test in browser or curl)

```
minikube service wisecow-service --url
```



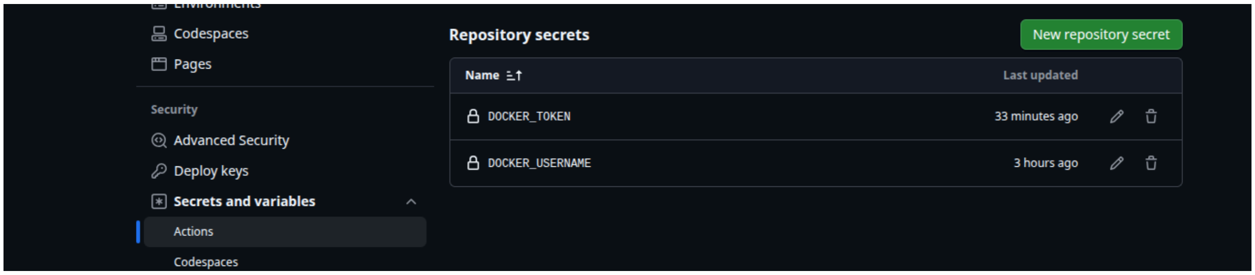
```
 ________________________________________
/ It were not best that we should all    \
| think alike; it is difference of       |
| opinion that makes horse-races. -- Mark |
\ Twain, "Pudd'nhead Wilson's Calendar"  /
 ----------------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

10. Create .github/workflows/cicd.yml by clicking on Actions → New workflow



11. Store Docker Hub credentials by going to Secrets and variables → Actions Create new secrets (DOCKERHUB_USERNAME &

DOCKERHUB_TOKEN)

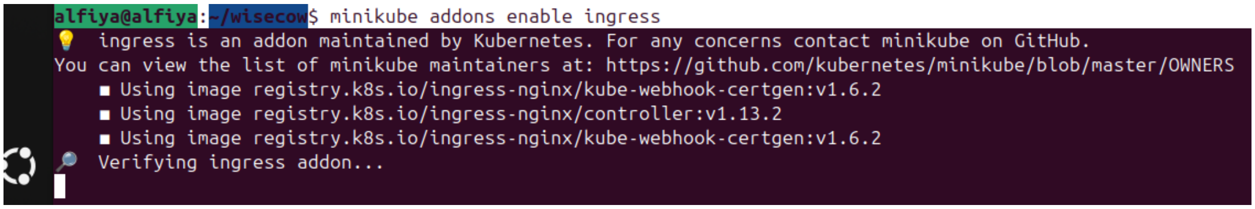


Repository secrets

Name	Last updated
DOCKER_TOKEN	33 minutes ago
DOCKER_USERNAME	3 hours ago

12. The workflow will automatically start on any new commits
13. Enable ingress in Minikube

```
minikube addons enable ingress
```



```
alfiya@alfiya:~/wisecow$ minikube addons enable ingress
💡  ingress is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
    ▪ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.6.2
    ▪ Using image registry.k8s.io/ingress-nginx/controller:v1.13.2
    ▪ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.6.2
🔎  Verifying ingress addon...
```

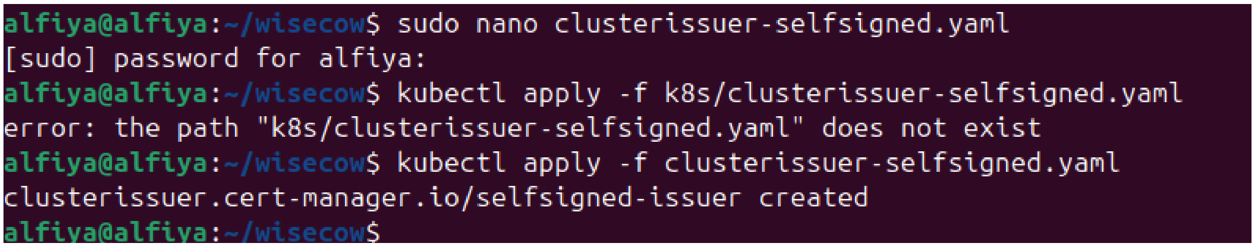
14. Install cert-manager

```
kubectl apply --validate=false -f
https://github.com/cert-manager/cert-manager/releases/lates
t/download/cert-manager.yaml

# wait until pods are ready
kubectl -n cert-manager rollout status deploy/cert-manager

kubectl -n cert-manager rollout status
deploy/cert-manager-webhook

kubectl -n cert-manager rollout status
deploy/cert-manager-cainjector
```

15. Create a self signed clusterIssuer
    - Create clusterissuer-selfsigned.yaml

```
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: selfsigned-issuer
spec:
  selfSigned: {}
```

- And apply kubectl apply -f k8s/clusterissuer-selfsigned.yaml

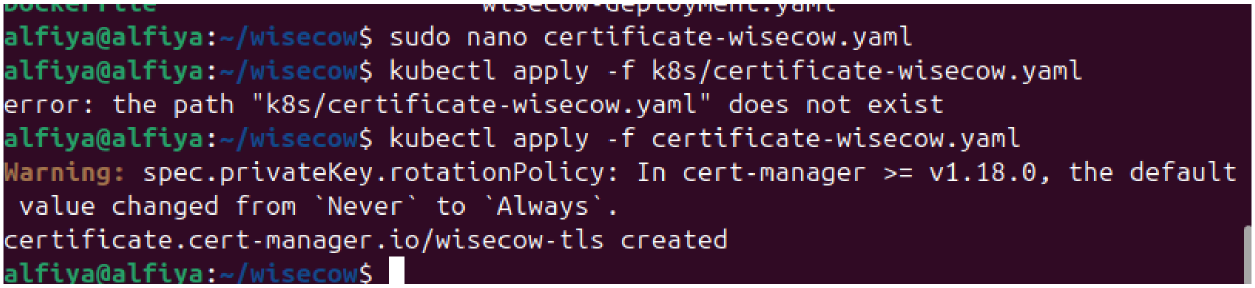


```
alfiya@alfiya:~/wisecow$ sudo nano clusterissuer-selfsigned.yaml
[sudo] password for alfiya:
alfiya@alfiya:~/wisecow$ kubectl apply -f k8s/clusterissuer-selfsigned.yaml
error: the path "k8s/clusterissuer-selfsigned.yaml" does not exist
alfiya@alfiya:~/wisecow$ kubectl apply -f clusterissuer-selfsigned.yaml
clusterissuer.cert-manager.io/selfsigned-issuer created
alfiya@alfiya:~/wisecow$
```

16.Certificate for Application

- Create a certificate-wisecow.yaml
- And apply `kubectl apply -f k8s/certificate-wisecow.yaml`

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: wisecow-tls
  namespace: default
spec:
  secretName: wisecow-tls-secret
  dnsNames:
    - wisecow.local   # or use nip.io like
wisecow.127.0.0.1.nip.io
  issuerRef:
    name: selfsigned-issuer
    kind: ClusterIssuer
```



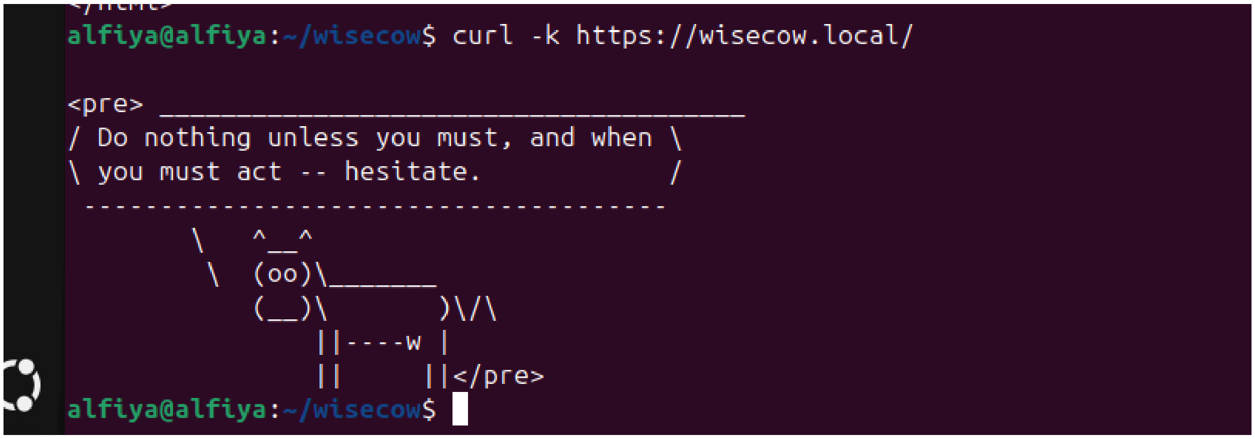
```
Dockerfile                    wisecow-deployment.yaml
alfiya@alfiya:~/wisecow$ sudo nano certificate-wisecow.yaml
alfiya@alfiya:~/wisecow$ kubectl apply -f k8s/certificate-wisecow.yaml
error: the path "k8s/certificate-wisecow.yaml" does not exist
alfiya@alfiya:~/wisecow$ kubectl apply -f certificate-wisecow.yaml
Warning: spec.privateKey.rotationPolicy: In cert-manager >= v1.18.0, the default
 value changed from `Never` to `Always`.
certificate.cert-manager.io/wisecow-tls created
alfiya@alfiya:~/wisecow$
```

17.Update the deployment and service file and apply both again

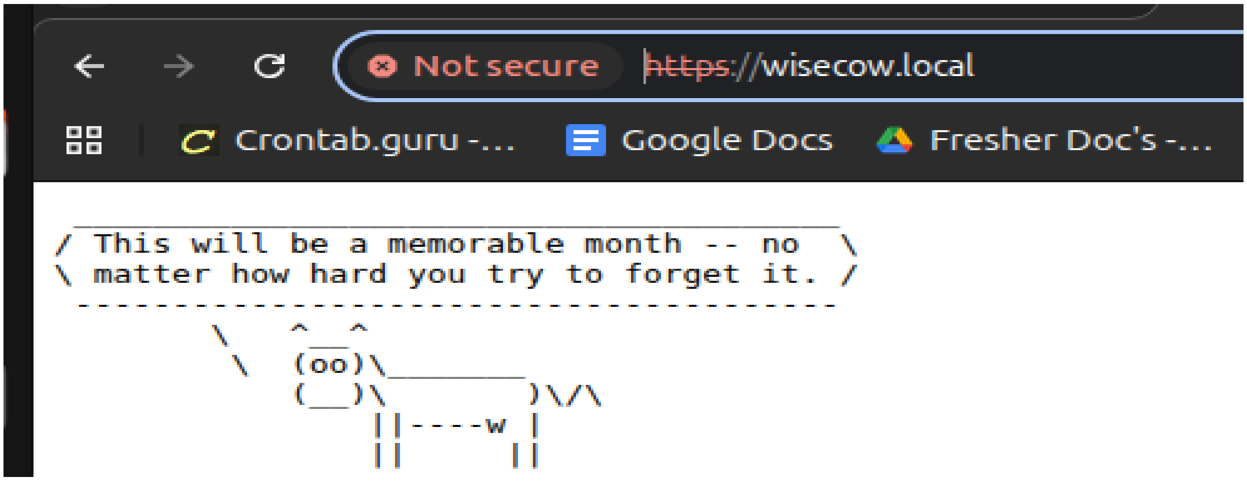
18. Create ingress file wisecow-ingress.yaml and apply

19. Add wisecow.local to /etc/hosts for host name mapping

20.Now test the HTTPS

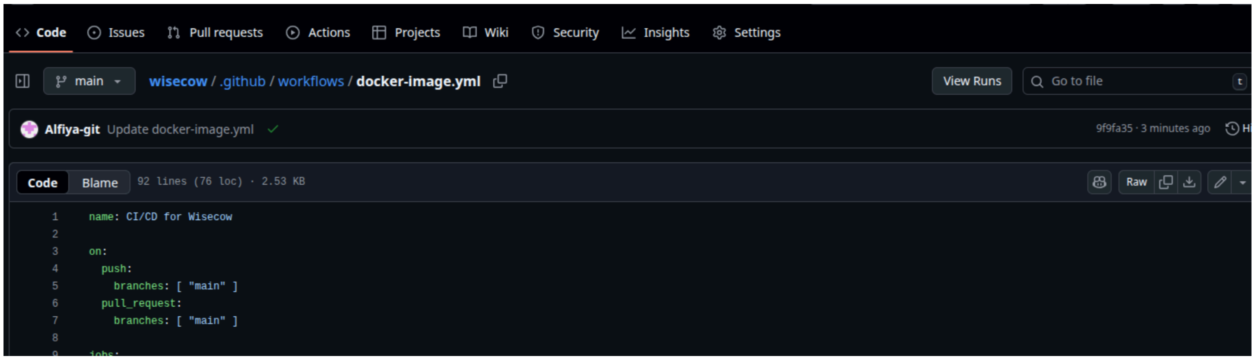


```
</html>
alfiya@alfiya:~/wisecow$ curl -k https://wisecow.local/

<pre> ______________________________________
/ Do nothing unless you must, and when \
\ you must act -- hesitate.            /
 --------------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||</pre>
alfiya@alfiya:~/wisecow$
```



```
 _________________________________________
/ This will be a memorable month -- no    \
\ matter how hard you try to forget it.   /
 -----------------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

21.Since tested locally apply it into the CI/CD script

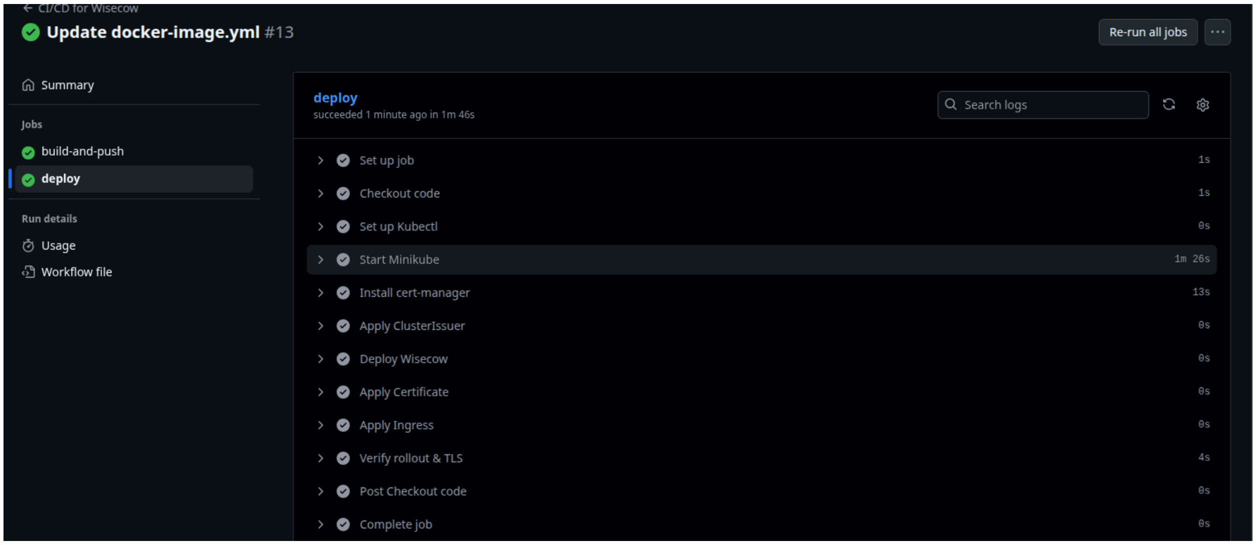


wisecow / .github / workflows / docker-image.yml

Alfiya-git Update docker-image.yml ✓ 9f9fa35 · 3 minutes ago

92 lines (76 loc) · 2.53 KB

```yaml
name: CI/CD for Wisecow

on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]

jobs:
```



Update docker-image.yml #13

Jobs: ✓ build-and-push ✓ deploy

deploy
succeeded 1 minute ago in 1m 46s

```
✓ Set up job                1s
✓ Checkout code             1s
✓ Set up Kubectl            0s
✓ Start Minikube        1m 26s
✓ Install cert-manager     13s
✓ Apply ClusterIssuer       0s
✓ Deploy Wisecow            0s
✓ Apply Certificate         0s
✓ Apply Ingress             0s
✓ Verify rollout & TLS      4s
✓ Post Checkout code        0s
✓ Complete job              0s
```