

```
docker build -t travel-memory-app .
```

## DEPLOY EVERYTHING

```
kubectl apply -f k8s/
```

Check:

```
kubectl get pods
```

```
kubectl get svc
```

```
kubectl get pvc
```

```
kubectl get ingress
```

```
----> Running in 95b49198d99d
----> Removed intermediate container 95b49198d99d
----> b620e504b23b
Successfully built b620e504b23b
Successfully tagged travel-memory-app:latest
ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app$ kubectl apply -f k8s/
configmap/travel-config created
deployment.apps/travel-app created
service/travel-app-service created
ingress.networking.k8s.io/travel-ingress created
deployment.apps/mongo created
persistentvolumeclaim/mongo-pvc created
service/mongo created
ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app$ kubectl get pods
kubectl get svc
kubectl get pvc
kubectl get ingress
NAME                                READY    STATUS              RESTARTS   AGE
microservice-1-d94466577-cn7hc     1/1      Running             1 (3h18m ago)  4h9m
microservice-2-f859c57bf-gp95r     1/1      Running             1 (3h18m ago)  4h9m
microservice-3-64465dc89c-69lx8    1/1      Running             1 (3h18m ago)  4h9m
mongo-796f87b8fc-xwnd8             1/1      Running             0           26s
travel-app-6d65c4ccb4-hmtn4        0/1      ImagePullBackOff    0           26s
travel-app-6d65c4ccb4-mfnf6        0/1      ImagePullBackOff    0           26s
NAME                                TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes                          ClusterIP  10.96.0.1        <none>            443/TCP       20h
microservice-1                      ClusterIP  10.103.237.229   <none>            80/TCP        4h9m
microservice-2                      ClusterIP  10.106.103.92    <none>            80/TCP        4h9m
microservice-3                      ClusterIP  10.99.253.152    <none>            80/TCP        4h9m
mongo                              ClusterIP  10.107.30.69     <none>            27017/TCP     26s
travel-app-service                  ClusterIP  10.106.31.26     <none>            80/TCP        26s
NAME      STATUS  VOLUME      CAPACITY  ACCESS  MODES  STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
mongo-pvc  Bound   pvc-82adb282-55a2-441c-b083-cface1c0e3bb  1Gi      RWO     standard      <unset>                 26s
NAME      CLASS  HOSTS
microservices-ingress  nginx  service1.local.com,service2.local.com,service3.local.com  192.168.49.2  80  4h9m
```

```

ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app$ kubectl get svc
NAME                TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes           ClusterIP   10.96.0.1        <none>            443/TCP          20h
microservice-1       ClusterIP   10.103.237.229   <none>            80/TCP           4h11m
microservice-2       ClusterIP   10.106.103.92    <none>            80/TCP           4h11m
microservice-3       ClusterIP   10.99.253.152    <none>            80/TCP           4h11m
mongo                ClusterIP   10.107.30.69     <none>            27017/TCP        2m37s
travel-app-service   ClusterIP   10.106.31.26     <none>            80/TCP           2m37s
ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app$ kubectl get pvc
NAME                STATUS      VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
mongo-pvc           Bound       pvc-82adb282-55a2-441c-b083-cfacelc0e3bb 1Gi         RWO            standard              <unset>                  2m46s
ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app$ kubectl get pods
NAME                READY     STATUS    RESTARTS   AGE
microservice-1-d94466577-cnrlc  1/1       Running   1 (3h21m ago)  4h11m
microservice-2-f859c57bf-gp95r  1/1       Running   1 (3h21m ago)  4h11m
microservice-3-64465dc89c-69lx8  1/1       Running   1 (3h21m ago)  4h11m
mongo-796f87b8fc-xwnd8          1/1       Running   0           3m4s
travel-app-6d65c4ccb4-hmtn4      0/1       ImagePullBackOff  0           3m4s
travel-app-6d65c4ccb4-mfnf6      0/1       ImagePullBackOff  0           3m4s
ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app$ kubectl get ingress
NAME                CLASS    HOSTS                                                                                                                                                                                                                                                                                                ADDRESS      PORTS   AGE
microservices-ingress  nginx    service1.local.com,service2.local.com,service3.local.com  192.168.49.2  80      4h11m
travel-ingress        nginx    memories.local.com                                         192.168.49.2  80      3m6s
ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app$

```

## Add to /etc/hosts

# Get Minikube IP

MINIKUBE\_IP=\$(minikube ip)

# Add to /etc/hosts

echo "\$MINIKUBE\_IP memories.local.com" | sudo tee -a /etc/hosts

# Verify

cat /etc/hosts | grep memories.local.com

```

ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app/k8s$ MINIKUBE_IP=$(minikube ip)
ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app/k8s$ echo "$MINIKUBE_IP memories.local.com" | sudo tee -a /etc/hosts
192.168.49.2 memories.local.com
ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app/k8s$ cat /etc/hosts | grep memories.local.com
192.168.49.2 memories.local.com
ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app/k8s$ curl -X POST http://memories.local.com/memories \
-H "Content-Type: application/json" \
-d '{
  "location": "Paris",
  "date": "2024-01-05",
  "description": "Eiffel Tower visit",
  "imageUrl": "https://example.com/img1.jpg"
}'
<html>
<head><title>503 Service Temporarily Unavailable</title></head>
<body>
<center><h1>503 Service Temporarily Unavailable</h1></center>
<hr><center>nginx</center>
</body>
</html>
ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app/k8s$

```

TEST API

CREATE

curl -X POST http://memories.local.com/memories \

-H "Content-Type: application/json" \

```
-d '{
  "location":"Paris",
  "date":"2024-01-05",
  "description":"Eiffel Tower visit",
  "imageUrl":"https://example.com/img1.jpg"
}'
```

GET

curl http://memories.local.com/memories

```
ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app$ sudo sh -c 'echo "192.168.49.2 memories.local.com" >> /etc/hosts'
ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app$ curl -X POST http://memories.local.com/memories \
-H "Content-Type: application/json" \
-d '{
  "location":"Paris",
  "date":"2024-01-05",
  "description":"Eiffel Tower visit",
  "imageUrl":"https://example.com/img1.jpg"
}'
{"location":"Paris","date":"2024-01-05","description":"Eiffel Tower visit","imageUrl":"https://example.com/img1.jpg","_id":"6937ca2ecd6888c3c8403006","__v":0}
ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app$ curl http://memories.local.com/memories
[{"_id":"6937c9a2c83de9d2fa22a9b8","location":"Paris","date":"2024-01-05","description":"Eiffel Tower visit","imageUrl":"https://example.com/img1.jpg","__v":0}, {"_id":"6937ca2ecd6888c3c8403006","location":"Paris","date":"2024-01-05","description":"Eiffel Tower visit","imageUrl":"https://example.com/img1.jpg","__v":0}]
ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app$
```

## AWS Deployment

```
ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app$ aws ecr get-login-password --region eu-west-2 | docker login --u
sername AWS --password-stdin 975050024946.dkr.ecr.eu-west-2.amazonaws.com

WARNING! Your credentials are stored unencrypted in '/home/ubuntu/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app$ docker tag adish786/travel-memory-app:latest 975050024946.dkr.ecr.eu-west-2.amazonaws.com/adish786/travel-memory-app:latest
ubuntu@ip-172-31-15-140:~/Container-Orchestration-Practice-Tasks/travel-memory-app$ docker push 975050024946.dkr.ecr.eu-west-2.amazonaws.com/adish786/travel-memory-app:latest
The push refers to repository [975050024946.dkr.ecr.eu-west-2.amazonaws.com/adish786/travel-memory-app]
5758fa576651: Pushed
040d39e0e149: Pushed
d5f514318493: Pushed
184f08b218e0: Pushed
82140d9a70a7: Pushed
f3b40b0cdeb1c: Pushed
0b1f26057bd0: Pushed
08000c18d16d: Pushed
latest: digest: sha256:f277acbf98bf4aec88bbaba577bfc369c7f89396c84fee2696fff4123d4c88a2c size: 1990
```

sudo docker pull mongo

sudo docker run -d -p 27017:27017 --name mongoddb -v /data/db:/data/db mongo

sudo docker exec -it mongoddb mongo

Amazon Elastic Container Service

Task definitions

adish-travel-memory-app

Revision 1

Containers

Amazon Elastic Container Service

Express Mode New

Clusters

Namespaces

Task definitions

Account settings

Amazon ECR [↗](#)

Repositories [↗](#)

AWS Batch [↗](#)

Documentation [↗](#)

Discontinue products [↗](#)

adish-travel-memory-app:1

Last updated  
December 9, 2025, 12:58 (UTC+5:30)

Deploy

Actions

Create new revision

Overview Info

ARN  
[am:aws:ecs:eu-west-2:975050024946:task-definition/adish-travel-memory-app:1](#)

Status  
ACTIVE

Time created  
December 9, 2025, 12:53 (UTC+5:30)

App environment  
Fargate

Task role  
-

Task execution role  
[ecsTaskExecutionRole](#) [↗](#)

Operating system/Architecture  
Linux/X86\_64

Network mode  
awsvpc

Fault injection  
Turned off

Containers

JSON

Task placement

Volumes (0)

Requires attributes

Tags

Task size

CloudShell

Feedback

Console Mobile App

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Amazon Elastic Container Service

Task definitions

adish-mongo

Revision 1

Containers

Amazon Elastic Container Service

Express Mode New

Clusters

Namespaces

Task definitions

Account settings

Amazon ECR [↗](#)

Repositories [↗](#)

AWS Batch [↗](#)

Documentation [↗](#)

Discontinue products [↗](#)

adish-mongo:1

Last updated  
December 9, 2025, 12:59 (UTC+5:30)

Deploy

Actions

Create new revision

Overview Info

ARN  
[am:aws:ecs:eu-west-2:975050024946:task-definition/adish-mongo:1](#)

Status  
ACTIVE

Time created  
December 9, 2025, 12:57 (UTC+5:30)

App environment  
Fargate

Task role  
-

Task execution role  
[ecsTaskExecutionRole](#) [↗](#)

Operating system/Architecture  
Linux/X86\_64

Network mode  
awsvpc

Fault injection  
Turned off

Containers

JSON

Task placement

Volumes (0)

Requires attributes

Tags

Task size

CloudShell

Feedback

Console Mobile App

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)