

Cloud Computing Assignment:CS GY 9223

New York University

Tandon School Of Engineering

Professor Sambit Sahu

Name:

- 1. Tanmay Anil Rathi (tr2452)**
- 2. Simardeep Singh Mehta (sm11377)**

Detailed Assignment Documentation

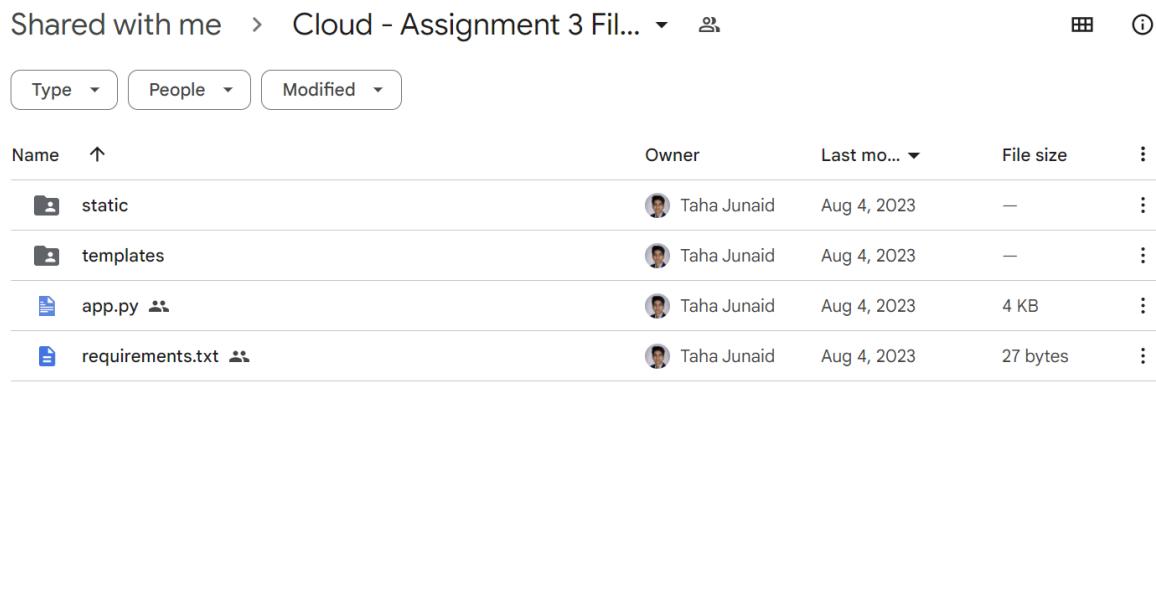
Part 1: Creating a Flask and MongoDB Application

Objective: Set up a simple To-Do application using Flask as the web framework and MongoDB for data persistence.

Steps:

1. Application Code Setup:

- Downloaded the provided Flask application source code.



Name	Owner	Last mo...	File size	⋮
static	Taha Junaid	Aug 4, 2023	—	⋮
templates	Taha Junaid	Aug 4, 2023	—	⋮
app.py	Taha Junaid	Aug 4, 2023	4 KB	⋮
requirements.txt	Taha Junaid	Aug 4, 2023	27 bytes	⋮

- Stored this locally and used WASP for running the commands.

- I have created different files for each step of this assignment.

```
(base) ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/Hw2/Source Code/Cloud - Assignment 3 Files (0.054s)
ls
Deployment.yaml      Deployment_Rolling.yaml  Services.yaml          app.py           flask-app-rc.yaml    requirements.txt   templates
Deployment_Health.yaml Dockerfile            alertmanager.yaml    docker-compose.yml  my-rule-file.yaml  static             test-pod.yaml

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/Hw2/Source Code/Cloud - Assignment 3 Files
```

- Brief description of each file:

1. Dockerfile: Contains the instructions for building the Docker image for your Flask application.

2. Deployment.yaml: This file defines the desired state of your Flask application deployment, including the number of replicas, container image, and labels for a Kubernetes deployment.
3. Services.yaml: This file defines the Kubernetes services that expose your Flask application and MongoDB to the network.
4. docker-compose.yml: Defines the multi-container Docker application setup, including services, networks, and volumes.
5. flask-app-rc.yaml: Defines a ReplicationController for your Flask application to manage the number of pod replicas.
6. Deployment_Rolling.yaml: Similar to Deployment.yaml but configured for a rolling update strategy, ensuring zero downtime during updates.
7. Deployment_Health.yaml: This file includes the deployment configuration along with health check probes (liveness and readiness) for your Flask application.
8. my-rule-file.yaml: A Prometheus configuration file that defines alert conditions and rules.
9. alertmanager.yaml: Configuration file for Alertmanager, which handles alerts sent by client applications such as Prometheus.
10. test-pod.yaml: For creating a temporary pod within Kubernetes, for testing the Prometheus service

Part 2: Containerizing the Application with Docker

Objective: Create Docker containers for the Flask application and MongoDB to allow for easy deployment and scaling.

Steps:

1. Docker Installation:

- Followed the official Docker installation guide to install Docker on the local machine.

2. Writing the Dockerfile for Flask:

- Created a `Dockerfile` which includes:
 - `FROM python:3.8-slim` to use a base Python image.
 - `WORKDIR /app` to set the working directory.
 - `COPY . /app` to copy the local code into the container.
 - `RUN pip install -r requirements.txt` to install dependencies.
 - `CMD ["python", "app.py"]` to specify the command to start the app.

The screenshot shows a code editor interface with several tabs open. The tabs include 'Deployment.yaml', 'Services.yaml', 'docker-compose.yaml', 'Dockerfile' (which is currently selected), 'app.py', and 'flask-app-rc.yaml'. The 'Dockerfile' tab contains the following code:

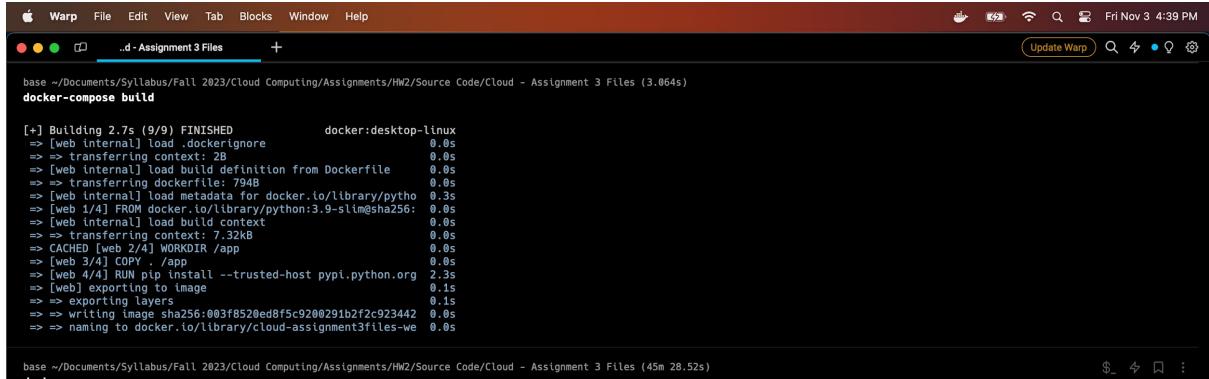
```
1 # Use an official Python runtime as a parent image
2 FROM python:3.9-slim
3
4 # Set the working directory to /app
5 WORKDIR /app
6
7 # Copy the current directory contents into the container at /app
8 COPY . /app
9
10 # Install any needed packages specified in requirements.txt
11 RUN pip install --trusted-host pypi.python.org -r requirements.txt
12
13 # Make port 5000 available to the world outside this container
14 EXPOSE 5000
15
16 # Define environment variable
17 ENV NAME World
18
19 # Run app.py when the container launches
20 CMD ["python", "app.py"]
```

The left sidebar shows a file tree with the following structure:

- ASSIGNMENTS
- HW1
- HW2
- Source Code
 - Cloud - Assignment 3 Files
 - static
 - templates
 - app.py
 - Deployment.yaml
 - docker-compose.yaml
 - Dockerfile
 - flask-app-rc.yaml
 - requirements.txt
 - Services.yaml
 - Screenshots
 - Homework Assignment 2.pdf

3. Building the Docker Image:

- Build the Flask Docker image using `docker compose build` and then use `docker compose up` for the image to be up and running.



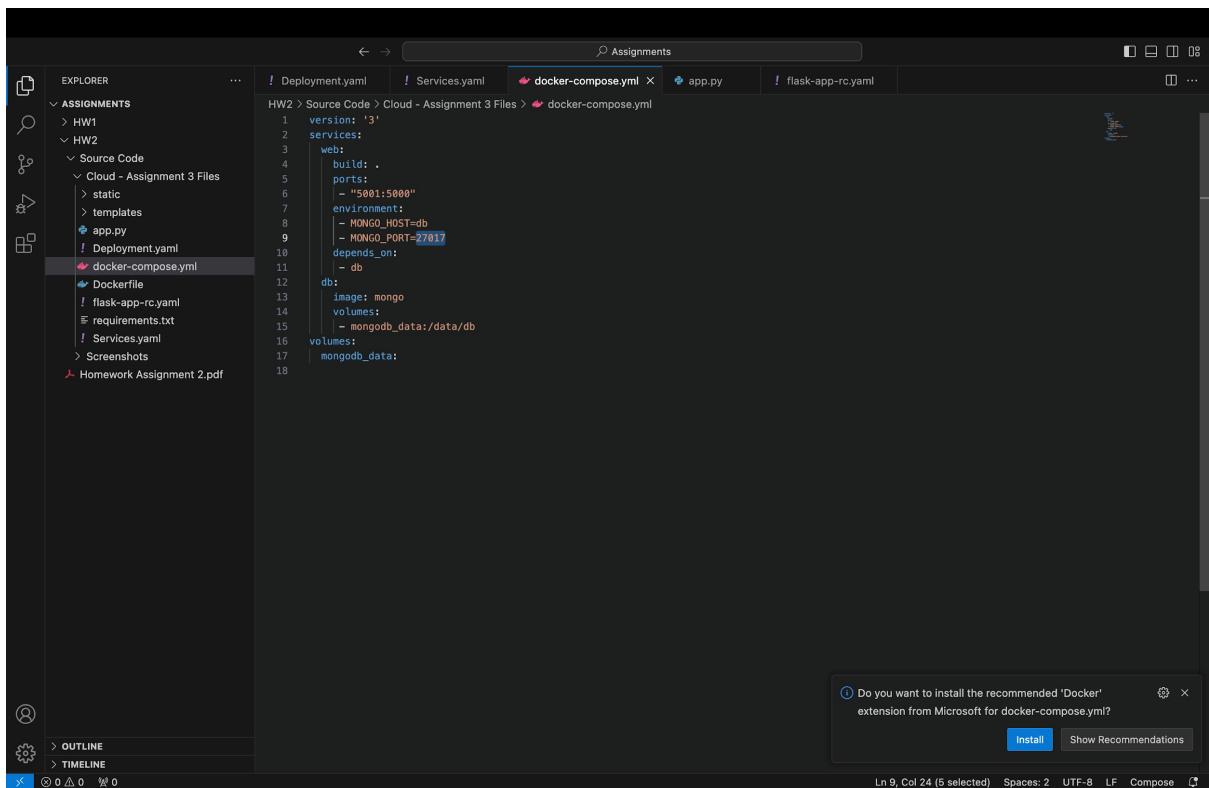
```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (3.064s)
docker-compose build

[+] Building 2.7s (9/9) FINISHED
   [web 1/9] load dockerignore          0.0s
--> <web internal> load build context  0.0s
--> <web internal> load build definition from Dockerfile  0.0s
--> <web internal> load metadata for docker.io/library/python: 0.0s
--> [web 1/4] FROM docker.io/library/python:3.9-slim@sha256: 0.0s
--> [web internal] load build context  0.0s
--> transferring context: 7.32kB      0.0s
--> CACHED [web 2/4] WORKDIR /app     0.0s
--> [web 3/4] COPY . /app             0.0s
--> [web 4/4] RUN pip install --trusted-host pypi.python.org 2.3s
--> [web] exporting to image          0.1s
--> exporting layers                0.1s
--> >> writing image sha256:003f8520ed8f5c9200291b2f2c923442 0.0s
--> >> naming to docker.io/library/cloud-assignment3files-we 0.0s

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (45m 28.52s)
```

4. Writing the docker-compose.yml:

- Defined services for the Flask app and MongoDB in `docker-compose.yml`.

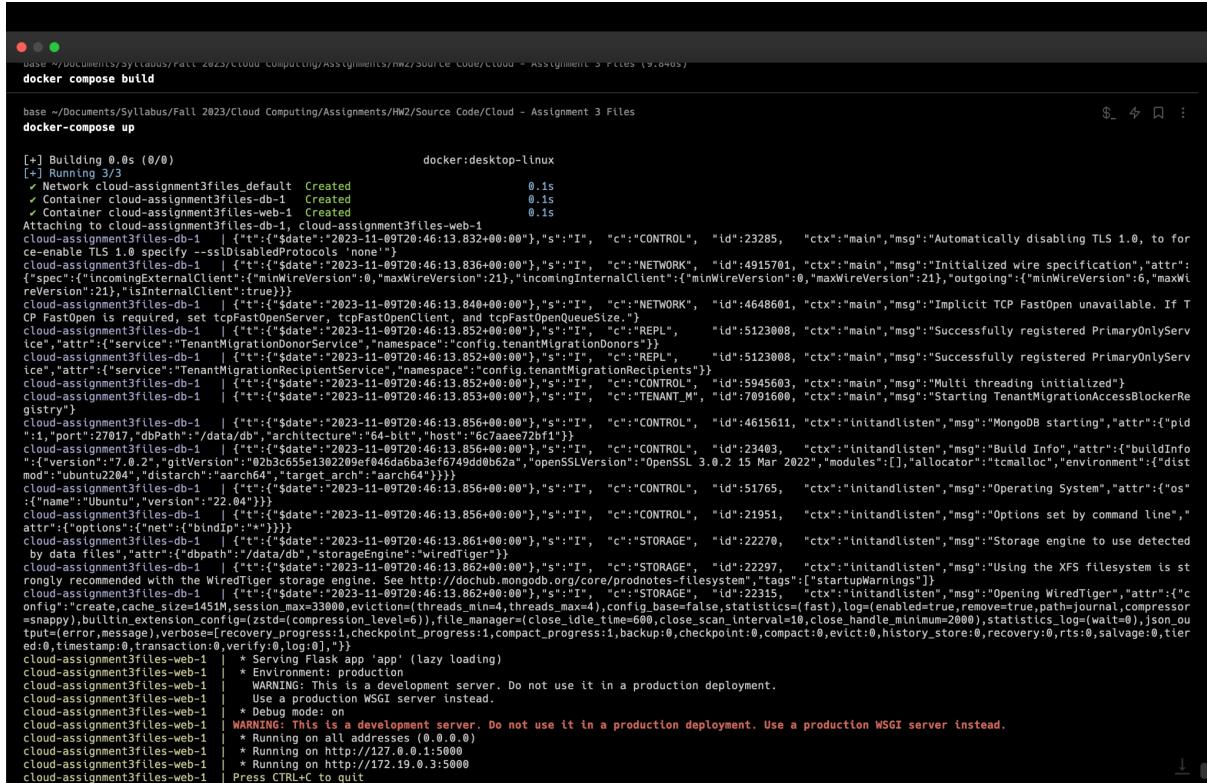


```
version: '3'
services:
  web:
    build: .
    ports:
      - "5001:5000"
    environment:
      - MONGO_HOST=db
      - MONGO_PORT=27017
    depends_on:
      - db
  db:
    image: mongo
    volumes:
      - mongodb_data:/data/db
volumes:
  mongodb_data:
```

A screenshot of the Visual Studio Code interface. The left sidebar shows a tree view of files and folders, including 'HW2', 'Source Code', 'Cloud - Assignment 3 Files', and several configuration files like 'Deployment.yaml', 'Services.yaml', 'docker-compose.yml', 'app.py', 'flask-app-rc.yaml', 'Dockerfile', 'requirements.txt', and 'Services.yaml'. The 'docker-compose.yml' file is currently selected and open in the main editor area. The code itself defines a 'web' service that builds from the current directory, maps port 5001 to 5000, and connects to a 'db' service. It also defines a 'db' service using the 'mongo' image and mounting a local volume for MongoDB data. A status bar at the bottom indicates 'Ln 9, Col 24 (5 selected) Spaces: 2 UTF-8 LF Compose'. A small modal window in the bottom right corner asks if the user wants to install the 'Docker' extension from Microsoft for 'docker-compose.yml'.

5. Local Testing with docker-compose:

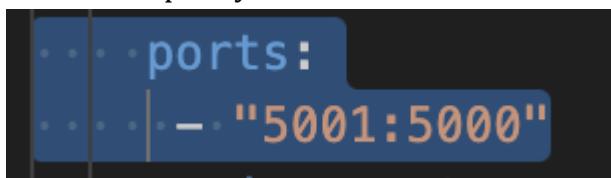
- Ran `docker-compose up` to start the multi-container application and tested its accessibility and functionality.



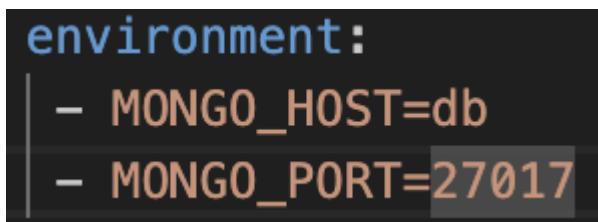
```
use ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (9.040s)
docker compose build
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files
$ docker-compose up

[+] Building 0.0s (0/0)
[+] Running 3/3
  Network cloud-assignment3files_default Created          0.1s
  Container cloud-assignment3files-db-1 Created          0.1s
  Container cloud-assignment3files-web-1 Created          0.1s
Attaching to cloud-assignment3files-db-1, cloud-assignment3files-web-1
cloud-assignment3files-db-1 | {"t": {"$date": "2023-11-09T20:46:13.832+00:00"}, "s": "I", "c": "CONTROL", "id": 23285, "ctx": "main", "msg": "Automatically disabling TLS 1.0, to for re-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
cloud-assignment3files-db-1 | {"t": {"$date": "2023-11-09T20:46:13.836+00:00"}, "s": "I", "c": "NETWORK", "id": 4915701, "ctx": "main", "msg": "Initialized wire specification", "attr": {"spec": {"incomingExternalClient": {"minWireVersion": 0, "maxWireVersion": 21}, "incomingInternalClient": {"minWireVersion": 0, "maxWireVersion": 21}, "outgoing": {"minWireVersion": 6, "maxWireVersion": 21}, "isInternalClient": true}}}
cloud-assignment3files-db-1 | {"t": {"$date": "2023-11-09T20:46:13.840+00:00"}, "s": "I", "c": "NETWORK", "id": 4648601, "ctx": "main", "msg": "Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize."}
cloud-assignment3files-db-1 | {"t": {"$date": "2023-11-09T20:46:13.852+00:00"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "TenantMigrationDonorService", "namespace": "config.tenantMigrationDonors"}}
cloud-assignment3files-db-1 | {"t": {"$date": "2023-11-09T20:46:13.852+00:00"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "TenantMigrationRecipientService", "namespace": "config.tenantMigrationRecipients"}}
cloud-assignment3files-db-1 | {"t": {"$date": "2023-11-09T20:46:13.852+00:00"}, "s": "I", "c": "CONTROL", "id": 5945603, "ctx": "main", "msg": "Multi threading initialized"}
cloud-assignment3files-db-1 | {"t": {"$date": "2023-11-09T20:46:13.852+00:00"}, "s": "I", "c": "TENANT_M", "id": 7091600, "ctx": "main", "msg": "Starting TenantMigrationAccessBlockerRegristration"
Cloud-assignment3files-db-1 | {"t": {"$date": "2023-11-09T20:46:13.856+00:00"}, "s": "I", "c": "CONTROL", "id": 4615611, "ctx": "initandlisten", "msg": "MongoDB starting", "attr": {"pid": "1", "port": "27017", "dbPath": "/data/db", "architecture": "64-bit", "host": "6cfaee72bf1"}}
cloud-assignment3files-db-1 | {"t": {"$date": "2023-11-09T20:46:13.856+00:00"}, "s": "I", "c": "CONTROL", "id": 23403, "ctx": "initandlisten", "msg": "Build Info", "attr": {"buildInfo": {"version": "7.0.2", "gitVersion": "02b3c655e130220ef046da6aba3ef6749dd0b62a", "openSSLVersion": "OpenSSL 3.0.2 15 Mar 2022", "modules": [], "allocator": "tcmalloc", "environment": {"distMod": "ubuntu2204", "distArch": "arch64", "targetArch": "arch64"}, "os": {"name": "Ubuntu", "version": "22.04"}}, "options": {"net": {"bindIp": ""}}}
cloud-assignment3files-db-1 | {"t": {"$date": "2023-11-09T20:46:13.856+00:00"}, "s": "I", "c": "CONTROL", "id": 21951, "ctx": "initandlisten", "msg": "Options set by command line", "attr": {"options": {"net": {"bindIp": ""}}}}
cloud-assignment3files-db-1 | {"t": {"$date": "2023-11-09T20:46:13.861+00:00"}, "s": "I", "c": "STORAGE", "id": 22270, "ctx": "initandlisten", "msg": "Storage engine to use detected by data files", "attr": {"dbpath": "/data/db", "storageEngine": "wiredTiger"}}
cloud-assignment3files-db-1 | {"t": {"$date": "2023-11-09T20:46:13.862+00:00"}, "s": "I", "c": "STORAGE", "id": 22297, "ctx": "initandlisten", "msg": "Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem#tags-[startupWarnings]}
cloud-assignment3files-db-1 | {"t": {"$date": "2023-11-09T20:46:13.862+00:00"}, "s": "I", "c": "STORAGE", "id": 22315, "ctx": "initandlisten", "msg": "Opening WiredTiger", "attr": {"config": "create,cache_size=451M,sessionDefault=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=fast,log=(enabled=true,remove=true,path=journal,compressor=none),journal=config,statistics_log=(wait=0),json_output=true,error_message=true,verbose=[recovery_progress=1,checkpoint_progress=1,compact_progress=1],compact=progress=1,backup=0,checkpoint=0,compact=0,evict=0,history_store=0,recovery=0,rts=0,salvage=0,tiered=0,timestamp=0,transaction=0,verify=0,log=0)"}
cloud-assignment3files-web-1 | * Serving Flask app 'app' (lazy loading)
cloud-assignment3files-web-1 | * Environment: production
cloud-assignment3files-web-1 | * WARNING: This is a development server. Do not use it in a production deployment.
cloud-assignment3files-web-1 | * Use a production WSGI server instead.
cloud-assignment3files-web-1 | * Debug mode: on
cloud-assignment3files-web-1 | * Environment: production
cloud-assignment3files-web-1 | * Running on all addresses (0.0.0.0)
cloud-assignment3files-web-1 | * Running on http://127.0.0.1:5000
cloud-assignment3files-web-1 | * Running on http://172.19.0.3:5000
cloud-assignment3files-web-1 | Press CTRL+C to quit
```

Accessed the website on port 5001 on my host machine as it was mapped in the docker-compose.yml file-



- The two env variables have also been set that the flask app.py expects





To-Do Reminder

ALL Uncompleted Completed About

Search Reference: Unique ID ▾ Search Task Search

To-Do LIST Version 2:

Status	Task Name	Description Name	Date	Priority	Remove	Modify
X	Tanmay taskd	sadsad		None		

Add a Task

Taskname Enter Description here...
mm/dd/yyyy

Priority ▾



After testing, I removed the containers by running this command-

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud -Assignment 3
docker compose down

[+] Running 3/3
 ✓ Container cloud-assignment3files-web-1  Removed          1.5s
 ✓ Container cloud-assignment3files-db-1  Removed          0.4s
 ✓ Network cloud-assignment3files_default  Removed          0.1s
```

6. Pushing Image to Docker Hub:

- Logged into Docker Hub via `docker login`.
- Tagged the local image to match the repository on Docker Hub using `docker tag`.
- Pushed the image using `docker push`.

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Fri Nov 3 4:35 PM

hub.docker.com/repository/docker/tr2452/my_todo_app/general

Google NYU Coursework Visa Excel Sheets Travel Internships Home | Microsoft... Tools News Shopping

Hackathon time! Join us for the Docker AI/ML Hackathon now through November 7th. Sign up now!

dockerhub Search Docker Hub Explore Repositories Organizations Help Upgrade tr2452

tr2452 / Repositories / my_todo_app / General Using 0 of 1 private repositories. Get more

General Tags Builds Collaborators Webhooks Settings

Add a short description for this repository The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results. Update

tr2452 / my_todo_app Docker commands Public View

Description This repository does not have a description.

Last pushed: 3 days ago

Docker commands To push a new tag to this repository:

```
docker push tr2452/my_todo_app:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
v1	Ubuntu	Image	17 hours ago	3 days ago

See all Go to Advanced Image Management

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](#).



Part 3: Deploying the Application on Minikube

Objective: Use Minikube to create a local Kubernetes cluster and deploy the application within.

Steps:

1. Minikube Installation:

- Installed Minikube by following the instructions from Kubernetes' official documentation.

2. Starting Minikube:

- Started the Minikube cluster with `minikube start`.

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (45.354s)
minikube start
  • Using image gcr.io/k8s-minikube/storage-provisioner:v5
  • Using image docker.io/kubernetesui/dashboard:v2.7.0
  ! Some dashboard features require the metrics-server addon. To enable all features please run:
    minikube addons enable metrics-server

  * Enabled addons: storage-provisioner, default-storageclass, dashboard
  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

Warp can notify you when long-running commands finish. Learn more Turn on notifications Don't show this again
```

We can check the context is generated and also that it is the current context that we are working with kubectl

	CLUSTER	NAME	NAMESPACE	AUTHINFO
*	minikube	minikube	arn:aws:eks:us-east-1:153437803668:cluster/my-eks-cluster	arn:aws:eks:us-east-1:153437803668:cluster/my-eks-cluster
		default		arn:aws:eks:us-east-1:153437803668:cluster/my-ek

3. Creating Kubernetes Manifests:

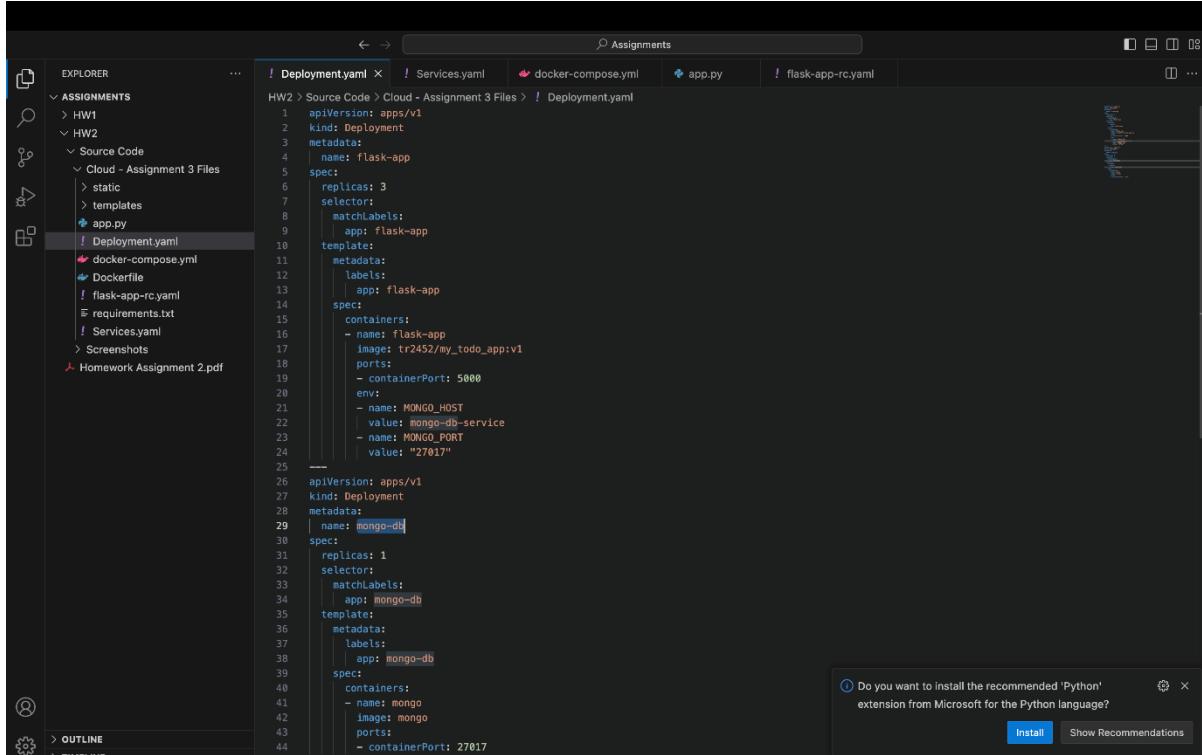
- Created deployment and service manifests for the Flask application and MongoDB.
- Defined resource requests and limits for the pods.
- The two env variables have also been set that the flask app.py expects

1. MONGO_HOST

Its value is set to the service name assigned to mongo-db-service

2. MONGO_PORT

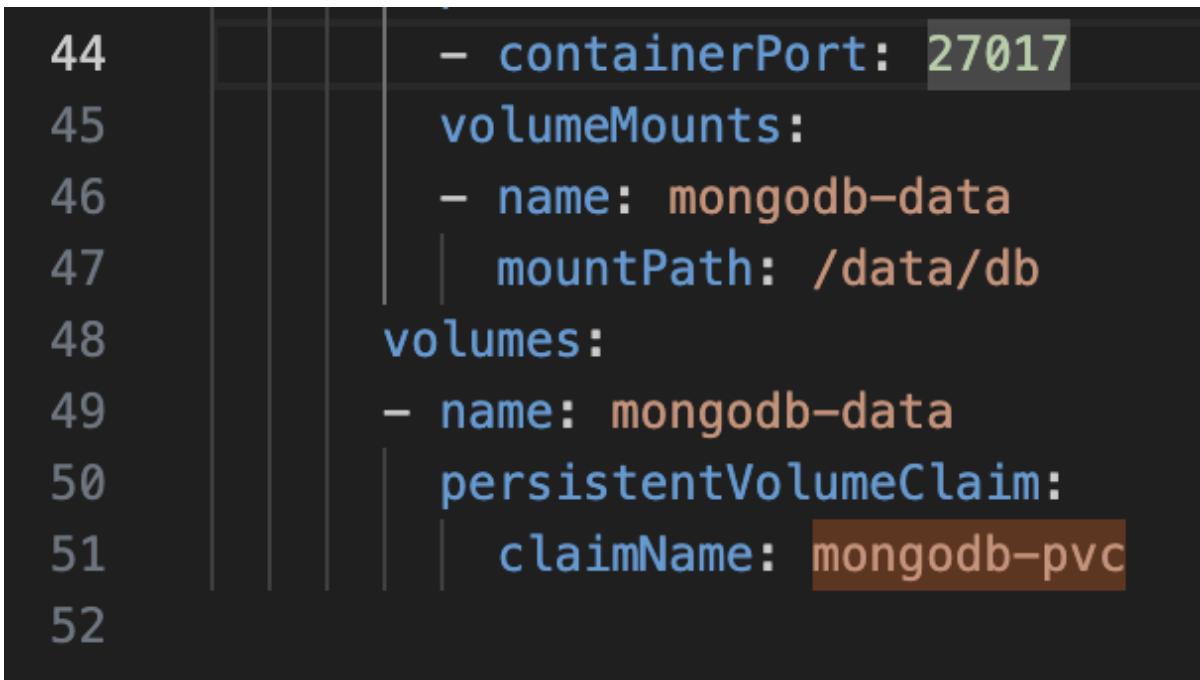
The app expects to listen on this port 27017.



```
! Deployment.yaml x ! Services.yaml docker-compose.yml app.py flask-app-rc.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: flask-app
5  spec:
6    replicas: 3
7    selector:
8      matchLabels:
9        app: flask-app
10   template:
11     metadata:
12       labels:
13         app: flask-app
14     spec:
15       containers:
16         - name: flask-app
17           image: tr2452/my_todo_app:v1
18           ports:
19             - containerPort: 5000
20             - name: MONGO_HOST
21               value: mongo-db-service
22             - name: MONGO_PORT
23               value: "27017"
24
25
26  apiVersion: apps/v1
27  kind: Deployment
28  metadata:
29    name: mongo-db
30  spec:
31    replicas: 1
32    selector:
33      matchLabels:
34        app: mongo-db
35    template:
36      metadata:
37        labels:
38          app: mongo-db
39    spec:
40      containers:
41        - name: mongo
42          image: mongo
43          ports:
44            - containerPort: 27017
```

A small pop-up window in the bottom right corner of the code editor asks if the user wants to install the 'Python' extension from Microsoft, with 'Install' and 'Show Recommendations' buttons.

- The MongoDB deployment part of your configuration requires changes to include the PersistentVolumeClaim (PVC)



```
44   - containerPort: 27017
45
46   volumeMounts:
47     - name: mongodb-data
48       mountPath: /data/db
49
50   volumes:
51     - name: mongodb-data
52       persistentVolumeClaim:
53         claimName: mongodb-pvc
```

- PVC in the volume mount (mongodb-pvc) matches the name specified in your PVC YAML file.

! Deployment.yaml ! mongodb-pvc.yaml X docker-compose.yml

! mongodb-pvc.yaml

```

1  apiVersion: v1
2  kind: PersistentVolumeClaim
3  metadata:
4    name: mongodb-pvc
5  spec:
6    accessModes:
7      - ReadWriteOnce
8    resources:
9      requests:
10     storage: 1Gi
11   storageClassName: gp2

```

- Created a service of type `LoadBalancer` to expose the Flask app to the host machine.

The screenshot shows a dark-themed code editor interface with several tabs open. On the left, there's an 'EXPLORER' sidebar showing project structure with folders like 'ASSIGNMENTS', 'HW1', 'HW2', and 'Source Code'. Under 'Source Code', there are files: 'Deployment.yaml', 'docker-compose.yml', 'Dockerfile', 'flask-app-rc.yaml', 'requirements.txt', and 'Services.yaml'. The 'Services.yaml' tab is currently active, displaying its contents:

```

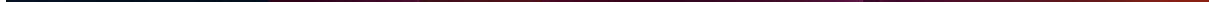
# Flask Service
apiVersion: v1
kind: Service
metadata:
  name: flask-mongo-service
spec:
  selector:
    app: flask-app
  ports:
    - protocol: TCP
      port: 5001
      targetPort: 5000
      type: LoadBalancer
  ---
# MongoDB Service
apiVersion: v1
kind: Service
metadata:
  name: mongo-db-service
spec:
  selector:
    app: mongo-db
  ports:
    - protocol: TCP
      port: 27017
      targetPort: 27017

```

At the bottom right, status information is visible: 'Ln 23, Col 10 (8 selected)' and file encoding 'UTF-8 LF YAML'.

4. Creating and Exposing Deployments:

- Applied the Kubernetes manifests using `kubectl apply -f <manifest>.yaml`.



```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.348s)
kubectl apply -f Deployment.yaml
deployment.apps/flask-app unchanged
deployment.apps/mongo-db unchanged

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.224s)
kubectl apply -f Services.yaml
service/flask-mongo-service unchanged
service/mongo-db-service unchanged

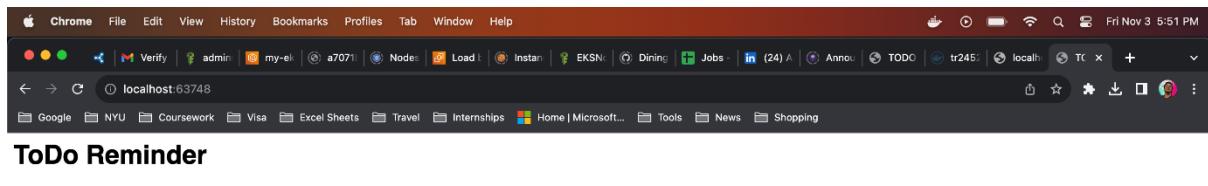
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files
```

5. Local Testing on Minikube:

- Retrieved the exposed URL using `kubectl get svc`.
- Tested the application to ensure that it was running correctly on the Minikube cluster.



```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files
minikube service flask-mongo-service --url
http://127.0.0.1:63748
! Because you are using a Docker driver on darwin, the terminal needs to be open to run it.
```



- `kubectl delete pod mongo-db-5d98c64fcf-5pnn2`

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3
kubectl get pods
  NAME           READY   STATUS    RESTARTS   AGE
flask-app-78b6764477-h2hvt   1/1     Running   4 (18m ago)   8d
flask-app-78b6764477-t464n   1/1     Running   4 (18m ago)   8d
flask-app-78b6764477-z72c5   1/1     Running   4 (18m ago)   8d
mongo-db-5d98c64fcf-5pnn2   1/1     Running   0          15m

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3
kubectl delete pod mongo-db-5d98c64fcf-5pnn2
pod "mongo-db-5d98c64fcf-5pnn2" deleted

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3
kubectl get pods
  NAME           READY   STATUS    RESTARTS   AGE
flask-app-78b6764477-h2hvt   1/1     Running   4 (18m ago)   8d
flask-app-78b6764477-t464n   1/1     Running   4 (18m ago)   8d
flask-app-78b6764477-z72c5   1/1     Running   4 (18m ago)   8d
mongo-db-5d98c64fcf-7n9x7   1/1     Running   0          9s
```

localhost:50460/list

[Java](#) [Presentation tools](#) [Placement](#) [ML, AI](#) [Tanmay](#) [CP](#) [GRE Prep](#) [Speedtest by Ookla...](#) [Courses to pick](#) [Unis Customization](#) [Blogs](#) [STONKS](#) [Visa](#)

ToDo Reminder

[ALL](#) [Uncompleted](#) [Completed](#) [About](#)

Search Reference:

To-Do LIST :							
Status	Task Name	Description Name	Date	Priority	Remove	Modify	
X	Task1	Description1	2023-11-12	Medium !!			
X	Task3	dsada	2023-11-01	Medium !!			

- After MongoDB restarted, I ensured the data was still there.

localhost:50460/list

[Java](#) [Presentation tools](#) [Placement](#) [ML, AI](#) [Tanmay](#) [CP](#) [GRE Prep](#) [Speedtest by Ookla...](#) [Courses to pick](#) [Unis Customization](#) [Blogs](#) [STONKS](#) [Visa](#)

ToDo Reminder

[ALL](#) [Uncompleted](#) [Completed](#) [About](#)

Search Reference:

To-Do LIST :							
Status	Task Name	Description Name	Date	Priority	Remove	Modify	
X	Task1	Description1	2023-11-12	Medium !!			
X	Task3	dsada	2023-11-01	Medium !!			

Part 4: Deploying the Application on AWS EKS

Objective: Set up AWS EKS to host the Kubernetes cluster and deploy the Flask application for production-like environment testing.

Steps:

1. AWS EKS Cluster Creation:

- Used the AWS Management Console to create an EKS cluster, specifying the node types and sizes.

The screenshot shows the AWS Management Console interface for the Amazon Elastic Kubernetes Service (EKS). On the left, there's a sidebar with 'Clusters' selected. The main area displays a table titled 'Clusters (1) info' with one entry: 'my-eks-cluster'. The table includes columns for 'Cluster name', 'Status', 'Kubernetes version', and 'Provider'. The status is 'Active', the version is '1.28', and the provider is 'EKS'.

- Created a node group with 2 EC2 instances or ARM Type

The screenshot shows the AWS Management Console interface for the Amazon Elastic Kubernetes Service (EKS). On the left, there's a sidebar with 'Clusters' selected. The main area has tabs for 'Compute', 'Networking', 'Add-ons', 'Authentication', 'Logging', 'Update history', and 'Tags', with 'Compute' currently selected. Under the 'Compute' tab, there are three sections: 'Nodes (2) Info', 'Node groups (1) Info', and 'Fargate profiles (0) Info'. The 'Nodes' section shows two nodes: 'ip-172-31-65-103.ec2.internal' and 'ip-172-31-90-87.ec2.internal', both of which are 'Ready'. The 'Node groups' section shows one group named 'my-cluster-arm64-nodegroup' with a size of 2, an AMI release version of '1.28.2-20231027', and a status of 'Active'.

- I faced an issue where when I selected the EC2 Instances of x86 architecture initially, it had trouble while running the python script.

-On further reading, I understood the issue was the difference in architecture of my image that was pushed on docker hub was of ARM Type (As it was created on my MAC) while the EC2 instances running the image were of type x86.

Once I selected the correct EC2 type of type ARM, it solved the issue.

- To this role, attached necessary permissions

The screenshot shows the AWS IAM console with the URL us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/roles/details/EKSNodeGroupRole?section=permissions. The page displays the details for the 'EKSNodeGroupRole' under the 'Identity and Access Management (IAM)' section. The 'Summary' tab is active, showing creation date (November 02, 2023, 21:07 UTC-04:00), ARN (arn:aws:iam::153437803668:role/EKSNodeGroupRole), and instance profile ARN (arn:aws:iam::153437803668:instance-profile/eks-20c5c9a0-7e87-a9ac-fd28-57a7d934722c). The 'Permissions' tab is selected, showing three managed policies attached: 'AmazonEC2ContainerRegistryReadOnly', 'AmazonEKS_CNI_Policy', and 'AmazonEKSWorkerNodePolicy'. The bottom of the screen shows the macOS dock with various application icons.

The screenshot shows the AWS CloudWatch Metrics interface. A line chart displays CPU usage percentage over a period of 1 hour. The Y-axis ranges from 0% to 100%, and the X-axis shows time intervals. The CPU usage fluctuates between 0% and 100% throughout the hour.

2. Configuring kubectl for EKS:

- Configured `kubectl` with AWS credentials and context to connect to the EKS cluster using `aws eks --region region-name update-kubeconfig --name cluster-name`.

The screenshot shows a Mac OS X desktop environment. The Dock at the bottom contains various application icons, and the system tray on the right shows network status, battery level, and other system information.

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.599s)
aws eks update-kubeconfig --name my-eks-cluster
Added new context arn:aws:eks:us-east-1:153437803668:cluster/my-eks-cluster to /Users/tanmayanilrathi/.kube/config

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.033s)
ls
Archive.zip          Dockerfile        app.py           requirements.txt    templates
Deployment.yaml      Services.yaml     docker-compose.yml  static
```

3. Deploying to EKS:

- Used the same Kubernetes manifests from Minikube but adjusted the image repository to point to Docker Hub.
- Applied the manifests to EKS using `kubectl apply`.

The screenshot shows a Mac OS X terminal window with several command-line operations for deploying a MongoDB instance to EKS.

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.094s)
kubectl config use-context arn:aws:eks:us-east-1:153437803668:cluster/my-eks-cluster
Switched to context "arn:aws:eks:us-east-1:153437803668:cluster/my-eks-cluster".

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.079s)
kubectl config get-contexts
CURRENT  NAME          CLUSTER          AUTHINFO
*        arn:aws:eks:us-east-1:153437803668:cluster/my-eks-cluster  arn:aws:eks:us-east-1:153437803668:cluster/my-eks-cluster  arn:aws:eks:us-east-1:153437803668:cluster/my-eks-cluster
  s-cluster
    minikube
    default

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (2.144s)
kubectl apply -f mongodb-pvc.yaml

persistentvolumeclaim/mongodb-pvc created
```

Resource types

- ▶ Workloads
- ▶ Cluster
- ▶ Service and networking
- ▶ Config and secrets
- ▼ Storage
 - PersistentVolumeClaims

Storage: PersistentVolumeClaims (1)

A request for storage by a user, similar to a Pod. [Learn more](#)

Name	Age
mongodb-pvc	Created 6 minutes ago

Resource types

- ▶ Workloads
- ▶ Cluster
- ▶ Service and networking
- ▶ Config and secrets
- ▼ Storage
 - PersistentVolumeClaims
 - PersistentVolumes

Storage: PersistentVolumes (1)

Persistent Volume is an API object that represents a piece of storage in the cluster. Available as a general, pluggable resource that persists beyond the lifecycle of any individual Pod.

Name	Age
pvc-056d04f9-5861-412b-a4b1-e9d225d80da2	Created 4 minutes ago

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.964s)
kubectl get pvc
NAME      STATUS    VOLUME          CAPACITY   ACCESS MODES   STORAGECLASS   AGE
mongodb-pvc  Bound    pvc-056d04f9-5861-412b-a4b1-e9d225d80da2  1Gi        RWO           gp2            86s

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.654s)
kubectl get pv
NAME          CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS    CLAIM           STORAGECLASS   REASON   AGE
pvc-056d04f9-5861-412b-a4b1-e9d225d80da2  1Gi        RWO           Delete       Bound    default/mongodb-pvc  gp2          38s
```

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (1.17s)
kubectl apply -f Deployment.yaml
deployment.apps/flask-app created
deployment.apps/mongo-db created

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.991s)
kubectl apply -f Services.yaml
service/flask-mongo-service created
service/mongo-db-service created
```

Amazon Elastic Kubernetes Service

Instances | EC2 | us-east-1 my-eks-cluster | Clusters | EBS

Overview Resources Compute Networking Add-ons Authentication Logging Update history Tags

Resource types

- ▶ Workloads
- ▶ Cluster
- ▼ Service and networking
 - Services
 - Endpoints
 - EndpointSlices
 - Ingresses
 - IngressClasses
- ▶ Config and secrets
- ▶ Storage
- ▶ Authentication
- ▶ Authorization
- ▶ Policy
- ▶ Extensions

Service and networking: Services (4)

Service is an abstract way to expose an application running on a set of Pods as a network service. [Learn more](#)

Name	Age
flask-mongo-service	Created November 3, 2023, 00:26 (UTC-04:00)
kube-dns	Created November 2, 2023, 20:12 (UTC-04:00)
kubernetes	Created November 2, 2023, 20:12 (UTC-04:00)
mongo-db-service	Created November 3, 2023, 00:26 (UTC-04:00)

5. Testing on EKS:

- Retrieved the public endpoint using `kubectl get svc`.

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.655s)
kubectl get svc
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP
flask-mongo-service   LoadBalancer  10.100.218.137  a7071b6343e6d4863beb6e3ebb36d2c0-1556485595.us-east-1.elb.amazonaws.com
kubernetes       ClusterIP    10.100.0.1     <none>
mongo-db-service  ClusterIP    10.100.128.214   <none>

PORT(S)          AGE
5001:30883/TCP  77s
443/TCP         24m
27017/TCP       76s

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.351s)
kubectl get svc flask-mongo-service
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP
flask-mongo-service   LoadBalancer  10.100.218.137  a7071b6343e6d4863beb6e3ebb36d2c0-1556485595.us-east-1.elb.amazonaws.com

PORT(S)          AGE
5001:30883/TCP  96s
```

- Accessed the application through the provided LoadBalancer URL to ensure it was functional in the EKS environment.

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (1.86s)
kubectl delete pod aws-node-9vd49 -n kube-system

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.496s)
kubectl get pods -n kube-system
NAME            READY  STATUS    RESTARTS   AGE
aws-node-wgktc  2/2    Running   0          73m
aws-node-x7rjq  2/2    Running   0          45s
coredns-58488c5db-fx69v  1/1    Running   0          73m
coredns-58488c5db-kjcwz  1/1    Running   0          73m
kube-proxy-7j9qw  1/1    Running   0          73m
kube-proxy-8sc12  1/1    Running   0          73m

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (3.147s)
kubectl get pods -n kube-system
NAME            READY  STATUS    RESTARTS   AGE
aws-node-wgktc  2/2    Running   0          73m
aws-node-x7rjq  2/2    Running   0          23s
coredns-58488c5db-fx69v  1/1    Running   0          73m
coredns-58488c5db-kjcwz  1/1    Running   0          73m
kube-proxy-7j9qw  1/1    Running   0          73m
kube-proxy-8sc12  1/1    Running   0          73m

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.627s)
kubectl get services
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP
flask-mongo-service   LoadBalancer  10.100.55.13  ac51a211671484079abbff4c59f2fc0e1-247441923.us-east-1.elb.amazonaws.com
kubernetes       ClusterIP    10.100.0.1     <none>
mongo-db-service  ClusterIP    10.100.51.73   <none>

PORT(S)          AGE
5001:31438/TCP  25m
443/TCP         4h39m
27017/TCP       25m

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (1.326s)
kubectl get pods
NAME            READY  STATUS    RESTARTS   AGE
flask-app-f75bcbb867-2cs59  1/1    Running   0          15h
flask-app-f75bcbb867-8968j  1/1    Running   0          15h
flask-app-f75bcbb867-8tmpo  1/1    Running   0          15h
mongo-db-7485d8fd6-6phxq  1/1    Running   0          15h
my-app-5b697fc7cd-xbqjp  1/1    Running   0          13h
```

- Deleted the mongo-db pod to check if the data is persisted using pvc defined

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.858s)
kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
alertmanager-my-pro-kube-prometheus-sta-alertmanager-0   2/2     Running   0          2d5h
failing-pod                           0/1     CrashLoopBackOff   1662 (3m36s ago)   5d22h
flask-app-f75bcb867-cpczm            1/1     Running   0          69m
flask-app-f75bcb867-lmz7z            1/1     Running   0          69m
flask-app-rc-2bm4g                  1/1     Running   0          69m
flask-app-rc-95vjc                  1/1     Running   0          8d
flask-app-rc-n6tsk                  1/1     Running   0          8d
flask-app-rc-t4g2w                  1/1     Running   0          8d
flask-app-rc-w6hg9                  1/1     Running   0          6d3h
flask-app-rc-zwzp9                  1/1     Running   0          6d17h
mongo-db-7b9bf7cf85-tf42x           1/1     Running   0          7m48s
my-app-5b697fc7cd-dngjc             1/1     Running   0          6d1h
my-pro-grafana-65fc586568-5cd6n    3/3     Running   1 (47h ago)   5d22h
my-pro-kube-prometheus-sta-operator-6bb7bccfb7-7z88q      1/1     Running   0          5d22h
my-pro-kube-state-metrics-85997cfbc4-lvk7f                1/1     Running   0          5d22h
my-pro-prometheus-node-exporter-kxmls                      1/1     Running   0          5d22h
my-pro-prometheus-node-exporter-vrgq7                      1/1     Running   0          5d22h
prometheus-my-pro-kube-prometheus-0                         2/2     Running   0          5d22h

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (1.95s)
kubectl delete pod -l app=mongo-db
pod "mongo-db-7b9bf7cf85-tf42x" deleted
```

To-Do LIST :							
Status	Task Name	Description	Name	Date	Priority	Remove	Modify
X	Task 1 AWS	dawd			None		
X	Task 4	AWS 4			None		

Data is persisting after the new mongo-pod is created.

Part 5: Replication Controller Feature

Objective: Ensure high availability by managing the desired state of the application using a replication controller.

Steps:

1. Creating Replication Controller Manifest:

- Wrote a `flask-app-rc.yaml` defining the desired number of replicas.

```
apiVersion: v1
kind: ReplicationController
metadata:
  name: flask-app-rc
  labels:
    app: flask-app
spec:
  replicas: 4
  selector:
    app: flask-app
  template:
    metadata:
      labels:
        app: flask-app
    spec:
      containers:
        - name: flask-app
          image: tr2452/my_todo_app:v1
          ports:
            - containerPort: 5000
          env:
            - name: MONGO_HOST
              value: mongo-db-service
            - name: MONGO_PORT
              value: "27017"
```

2. Replica Management:

- Deployed the replication controller with `kubectl create -f flask-app-rc.yaml`.
- Verified that the specified number of replicas were running using `kubectl get pods`.

A screenshot of a Mac OS X terminal window titled ".d - Assignment 3 Files". The window shows several command-line sessions:

- The first session runs `kubectl get pods -l app=flask-app` and lists three pods: `flask-app-f75bcb867-2cs59`, `flask-app-f75bcb867-8tmp`, and `flask-app-f75bcb867-qg944`, all in a "Running" state with 0 restarts and an age of 15h.
- The second session runs `kubectl apply -f flask-app-rc.yaml` and outputs "replicationcontroller/flask-app-rc created".
- The third session runs `kubectl get rc` and shows a single replication controller named "flask-app-rc" with 5 desired and 5 current pods, each in a "Running" state with an age of 3s.
- The fourth session runs `kubectl get pods` and lists 11 pods across various namespaces, including `flask-app-f75bcb867-2cs59` through `my-app-5b697fc7cd-xbjp`.
- The fifth session runs `kubectl get pods -l app=flask-app` again, showing the same three pods as before.
- The sixth session runs `kubectl delete pod flask-app-rc-cgh6` and deletes the pod named "flask-app-rc-cgh6".
- The seventh session runs `kubectl get pods -l app=flask-app` again, showing the remaining two pods.

3. Testing Resiliency:

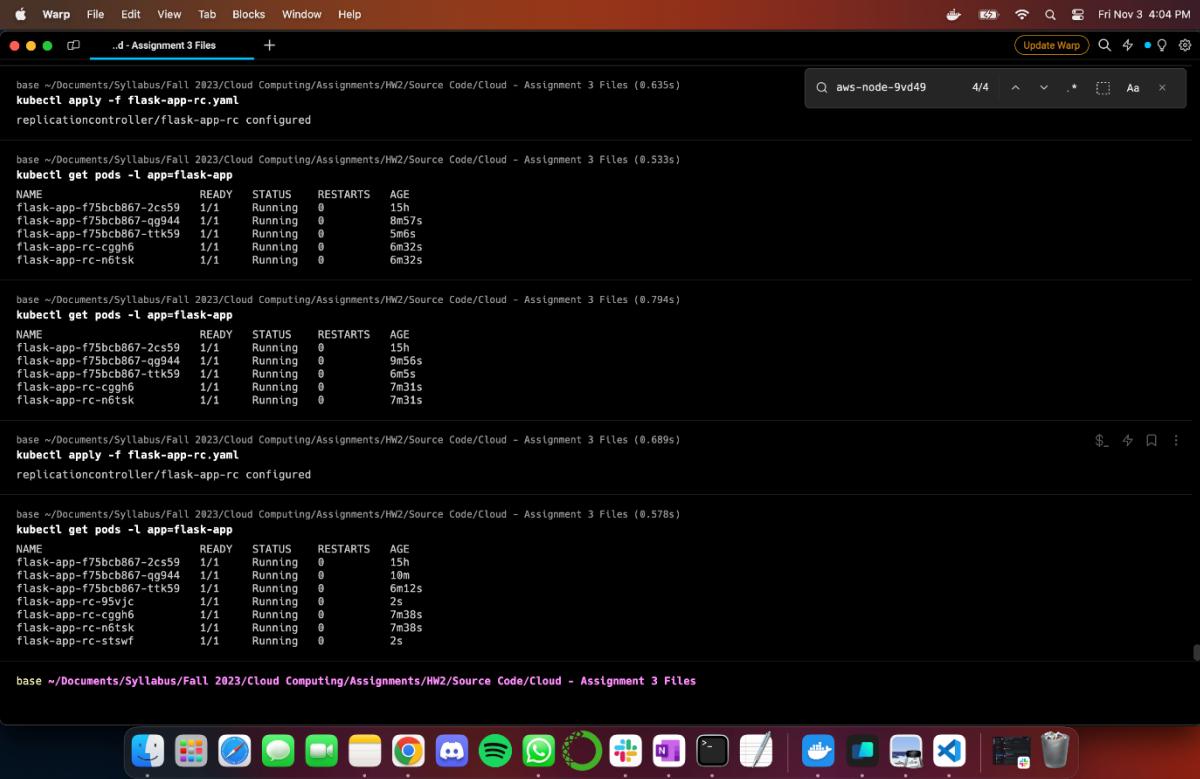
- Simulated failure by deleting a pod.
- Observed the replication controller creating a new pod to maintain the desired state.

A screenshot of a Mac OS X terminal window titled ".d - Assignment 3 Files". The window shows several command-line sessions:

- The first session runs `kubectl get pods -l app=flask-app` and lists five pods: `flask-app-f75bcb867-95vjc`, `flask-app-rc-cgh6`, `flask-app-rc-n6tsk`, and `flask-app-rc-stswf`, all in a "Running" state with 0 restarts and an age of 2s.
- The second session runs `kubectl delete pod flask-app-rc-cgh6` and deletes the pod named "flask-app-rc-cgh6".
- The third session runs `kubectl get pods -l app=flask-app` again, showing the remaining four pods.
- The fourth session runs `kubectl delete pod flask-app-rc-stswf` and deletes the pod named "flask-app-rc-stswf".
- The fifth session runs `kubectl get pods -l app=flask-app` again, showing the remaining three pods.
- The sixth session runs `kubectl get pods -l app=flask-app` again, showing the same three pods as before.

4. Scaling Replicas:

- Updated the number of replicas in the `replicationcontroller.yaml`.
- Applied the changes with `kubectl apply -f replicationcontroller.yaml` and verified the scaling action.



The screenshot shows a terminal window titled "Warp" with the following command history:

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/Hw2/Source Code/Cloud - Assignment 3 Files (0.635s)
kubectl apply -f flask-app-rc.yaml
replicationcontroller/flask-app-rc configured

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/Hw2/Source Code/Cloud - Assignment 3 Files (0.533s)
kubectl get pods -l app=flask-app
NAME READY STATUS RESTARTS AGE
flask-app-f75ccb867-2cs59 1/1 Running 0 15h
flask-app-f75ccb867-qg944 1/1 Running 0 8m5s
flask-app-f75ccb867-ttk59 1/1 Running 0 5m6s
flask-app-rc-cqgh6 1/1 Running 0 6m32s
flask-app-rc-n6tsk 1/1 Running 0 6m32s

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/Hw2/Source Code/Cloud - Assignment 3 Files (0.794s)
kubectl get pods -l app=flask-app
NAME READY STATUS RESTARTS AGE
flask-app-f75ccb867-2cs59 1/1 Running 0 15h
flask-app-f75ccb867-qg944 1/1 Running 0 9m56s
flask-app-f75ccb867-ttk59 1/1 Running 0 6m5s
flask-app-rc-cqgh6 1/1 Running 0 7m31s
flask-app-rc-n6tsk 1/1 Running 0 7m31s

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/Hw2/Source Code/Cloud - Assignment 3 Files (0.689s)
kubectl apply -f flask-app-rc.yaml
replicationcontroller/flask-app-rc configured

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/Hw2/Source Code/Cloud - Assignment 3 Files (0.578s)
kubectl get pods -l app=flask-app
NAME READY STATUS RESTARTS AGE
flask-app-f75ccb867-2cs59 1/1 Running 0 15h
flask-app-f75ccb867-qg944 1/1 Running 0 10m
flask-app-f75ccb867-ttk59 1/1 Running 0 6m12s
flask-app-rc-95vjc 1/1 Running 0 2s
flask-app-rc-cqgh6 1/1 Running 0 7m38s
flask-app-rc-n6tsk 1/1 Running 0 7m38s
flask-app-rc-stswf 1/1 Running 0 2s

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/Hw2/Source Code/Cloud - Assignment 3 Files
```

The terminal window is part of the Warp application, which is a terminal emulator for macOS. The window title is ".d - Assignment 3 Files". The status bar at the top right shows "Fri Nov 3 4:04 PM". The dock at the bottom of the screen contains icons for various Mac applications like Finder, Mail, Safari, and Visual Studio Code.

5. Displaying the creation of replication controller on EKS AWS::

- Showing the creation of flask-app-rc which is basically the replication controller
- Showing creation of pods in AWS management console.

The screenshot shows two consecutive views of the AWS Management Console for an Amazon Elastic Kubernetes Service (EKS) cluster named "my-eks-cluster".

Top Screenshot: The user is viewing the details of a specific pod named "flask-app-rc-2bm4g". The "Info" section displays the following details:

Status Running	Created November 3, 2023, 16:05 (UTC-04:00)	Namespace default
Controlled by ReplicationController/flask-app-rc	Last transition time November 3, 2023, 16:05 (UTC-04:00)	Node ip-172-31-65-103.ec2.internal
Pod IP 172.31.79.160		

The "Containers" section shows one container named "flask-app". The "Labels" and "Annotations" sections are currently empty.

Bottom Screenshot: The user has navigated to the "Pods" tab under the "Cluster Resources" section. The "Workloads: Pods (15)" table lists the following pods:

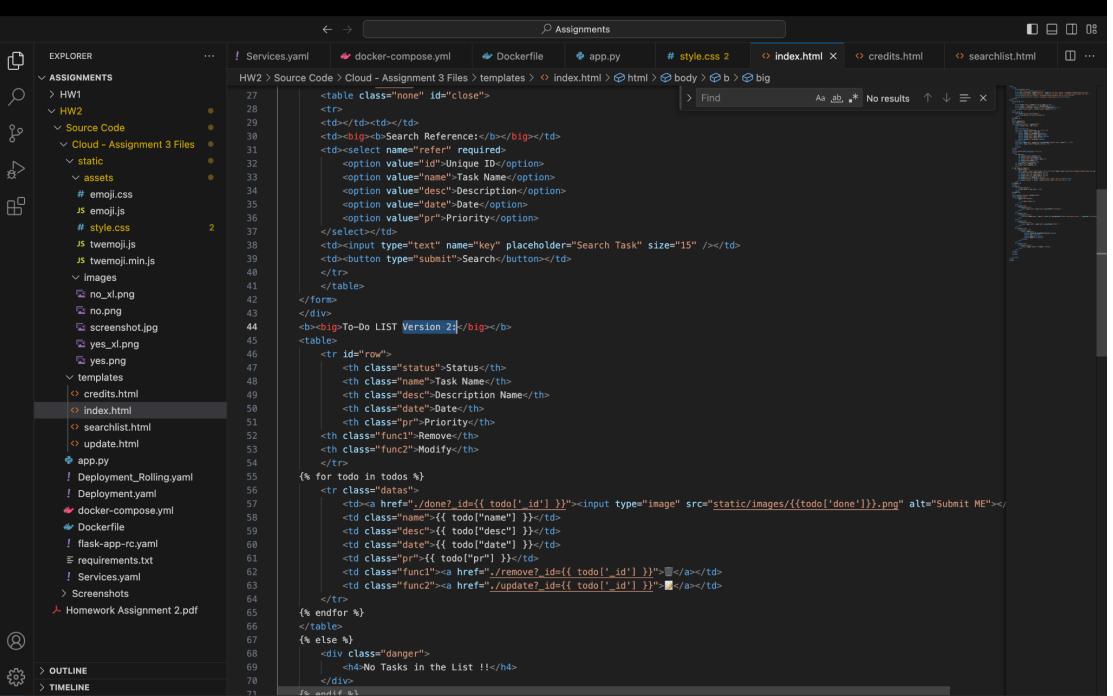
Name	Age
coredns-58488c5db-kjcwz	November 2, 2023, 23:37 (UTC-04:00)
flask-app-f75bcb867-2cs59	November 3, 2023, 00:22 (UTC-04:00)
flask-app-f75bcb867-qg944	a day ago
flask-app-f75bcb867-ttk59	a day ago
flask-app-rc-2bm4g	a day ago
flask-app-rc-95vjc	a day ago
flask-app-rc-n6tsk	a day ago

Part 6: Rolling Update Strategy

Objective: Implement zero-downtime deployments using a rolling update strategy.

Configure the Kubernetes Deployment to Use a Rolling Update Strategy:

-Making changes to the index.html file for the rolling update strategy to be implemented.



```
HW2 > Source Code > Cloud - Assignment 3 Files > templates > index.html > index.html > body > b > b > b
HW2 > Services.yaml docker-compose.yml Dockerfile app.py # style.css 2 credits.html searchlist.html ...
EXPLORER ... HW2 > ASSIGNMENTS > HW1 > HW2 > Source Code > Cloud - Assignment 3 Files > static > assets > emoji.css emoji.js style.css twemoji.js twemoji.min.js images no_xl.png no.png screenshot.jpg yes_xl.png yes.png < templates > credits.html < index.html > searchlist.html < update.html > app.py Deployment_Rolling.yaml Deployment.yaml docker-compose.yaml Dockerfile flask-app-rc.yaml requirements.txt Services.yaml Screenshots Homework Assignment 2.pdf ...
OUTLINE ... TIMELINE ...
```

```
<table>
  <tr><td><input type="text" name="key" placeholder="Search Task" size="15" /></td>
  <td><button type="submit">Search</button></td>
</tr>
</table>
<form>
  <div><b>To-Do LIST Version 2</b></div>
  <table border="1">
    <thead>
      <tr id="row1">
        <th class="status">Status</th>
        <th class="name">Task Name</th>
        <th class="desc">Description Name</th>
        <th class="date">Date</th>
        <th class="pr">Priority</th>
        <th class="func1">Remove</th>
        <th class="func2">Modify</th>
      </tr>
    <% for todo in todos %>
      <tr class="data">
        <td><a href="/done? id={{ todo['id'] }}></a><input type="image" src="static/images/{{todo['done']}}.png" alt="Submit ME"></td>
        <td>{{ todo['name'] }}</td>
        <td>{{ todo['desc'] }}</td>
        <td>{{ todo['date'] }}</td>
        <td>{{ todo['pr'] }}</td>
        <td><a href="/remove? id={{ todo['id'] }}></a></td>
        <td><a href="/update? id={{ todo['id'] }}></a></td>
      </tr>
    <% endfor %>
  </table>
  <% else %>
    <div class="danger">
      | <h4>No Tasks in the List !!</h4>
    </div>
  <% endelse %>
</div>
```

- Current state of the flask app deployed based on image v1

-Setting the Update Strategy to Rolling Update and Specifying the Maximum Number of Pods That Can Be Unavailable through parameters such as maxUnavailable and maxSurge to control the update process.

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask-app
spec:
  replicas: 3
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
  selector:
    matchLabels:
      app: flask-app
  template:
    metadata:
      labels:
        app: flask-app
    spec:
      containers:
        - name: flask-app
          image: tr2452/my_todo_app:v2
          ports:
            - containerPort: 5000
          env:
            - name: MONGO_HOST
              value: mongo-db-service
            - name: MONGO_PORT
              value: "27017"

```

Update the Docker Image for the Deployment to a New Version:

-updating the docker image in the deployment configuration file to point to the new version of the Docker image v2.

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask-app
spec:
  replicas: 3
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
  selector:
    matchLabels:
      app: flask-app
  template:
    metadata:
      labels:
        app: flask-app
    spec:
      containers:
        - name: flask-app
          image: tr2452/my_todo_app:v2
          ports:
            - containerPort: 5000
          env:
            - name: MONGO_HOST
              value: mongo-db-service
            - name: MONGO_PORT
              value: "27017"

```

-Building the new docker image based on the the new docker image v2

```

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/Hw2/Source Code/Cloud - Assignment 3 Files (1.087s)
docker build -t tr2452/my_todo_app:v2 .

[+] Building 0.4s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 796B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/python:3.9-slim
=> [1/4] FROM docker.io/library/python:3.9-slim@sha256:2b76740d0b7b94c29986855
=> [internal] load build context
=> => transferring context: 4.42kB
=> CACHED [2/4] WORKDIR /app
=> CACHED [3/4] COPY . /app
=> CACHED [4/4] RUN pip install --trusted-host pypi.python.org -r requirements
=> exporting to image
=> => exporting layers
=> => writing image sha256:b09f3e9794e9a57dabfc5b3d6bc92ea99b89e5e8302e642f765
=> => naming to docker.io/tr2452/my_todo_app:v2

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview

```

```

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/Hw2/Source Code/Cloud - Assignment 3 Files (0.217s)
docker images
REPOSITORY          TAG      IMAGE ID   CREATED          SIZE
tr2452/my_todo_app  v2      b09f3e9794e9  56 seconds ago  168MB
cloud-assignment3files-web  latest  92ec781bc148  56 seconds ago  168MB
<none>              <none>  e00e2fb7fb5c  28 minutes ago  168MB
<none>              <none>  731b2ef5ed66  23 hours ago   168MB
<none>              <none>  003f8520ed8f  2 days ago    168MB
<none>              <none>  20b9a5a3484f  2 days ago    168MB
<none>              <none>  16f3a2b81408  2 days ago    168MB
<none>              <none>  3f30123b6418  2 days ago    168MB
<none>              <none>  4e021e45256e  2 days ago    168MB
tr2452/my_todo_app  v1      a2971dee197d  3 days ago    168MB
<none>              <none>  7934a0f0e412  3 days ago    168MB
<none>              <none>  2087da445d30  3 days ago    168MB
<none>              <none>  6e60e43bd29  3 days ago    168MB
<none>              <none>  e821367305c9  3 days ago    168MB
<none>              <none>  cf7820090bdd  3 days ago    168MB
<none>              <none>  8a7f57f0e61f  3 days ago    168MB
<none>              <none>  6f1a2a005658  3 days ago    168MB
<none>              <none>  69b29d0ac006  3 days ago    167MB
mongo               latest  8b10e7ef0208  3 weeks ago   712MB
gcr.io/k8s-minikube/kicbase  v0.0.40  f52519afe5f6  3 months ago  1.1GB
hello-world         latest  b038788ddbb22  6 months ago  9.14kB

```

-Pushing the docker image to docker hub using 'docker push' command

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (8.443s)
docker push tr2452/my_todo_app:v2
The push refers to repository [docker.io/tr2452/my_todo_app]
9491912c72c9: Pushed
c4afda06d665: Pushed
d4ada40fe9c7: Pushed
45f43d445050: Mounted from library/python
d563d38f0e29: Mounted from library/python
7fff746fe2f74: Mounted from library/python
eb93fe2fd3a9: Mounted from library/python
70e628269d9f: Mounted from library/python
v2: digest: sha256:dc3544d683cc127e2e48210caac2f4d84231af7f6662e17e9c5dde9c836375ef size: 1997

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files
```

-pushed image on Dockerhub

 **tr2452 / my_todo_app**

Description

This repository does not have a description 

 Last pushed: a few seconds ago

Tags

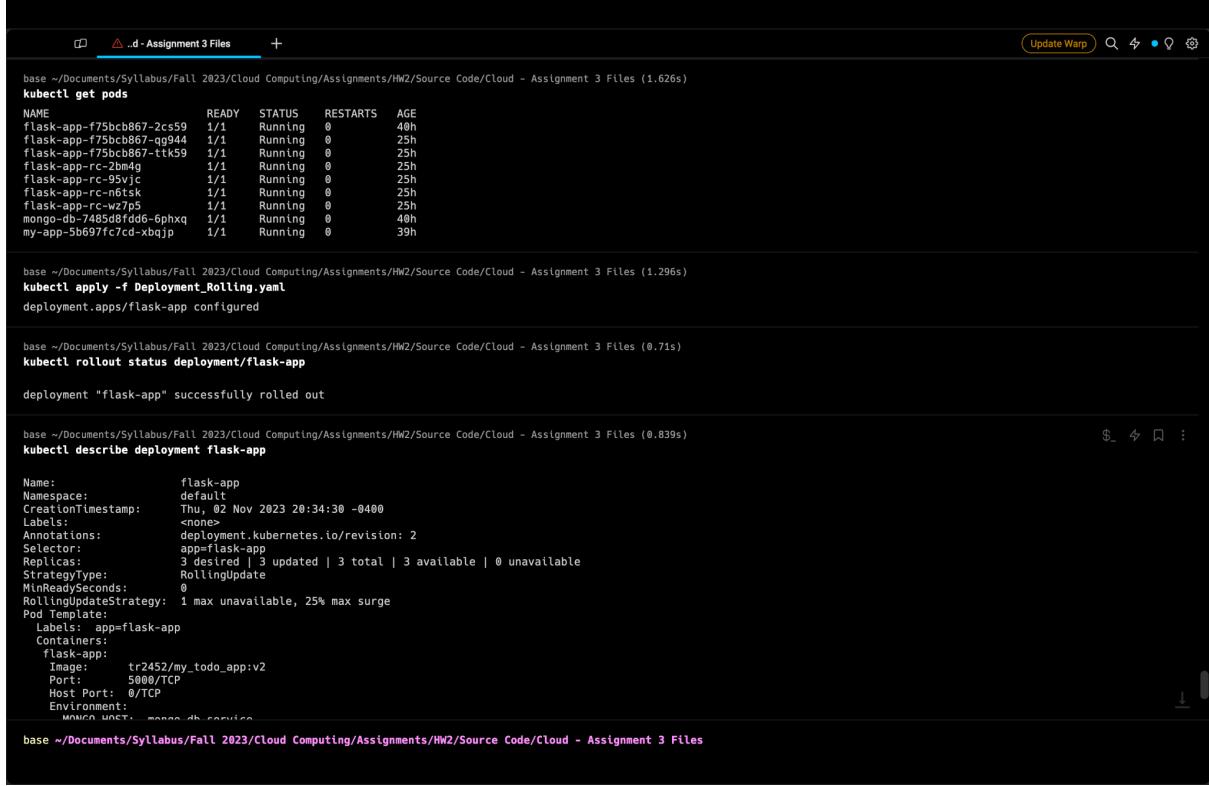
This repository contains 2 tag(s).

Tag	OS	Type	Pulled	Pushed
 v2		Image	---	a minute ago
 v1		Image	2 days ago	4 days ago

[See all](#) [Go to Advanced Image Management](#)

Trigger the Rolling Update by Updating the Deployment With the New Docker Image Version:

Applying the updated deployment configuration to your cluster using 'kubectl apply'



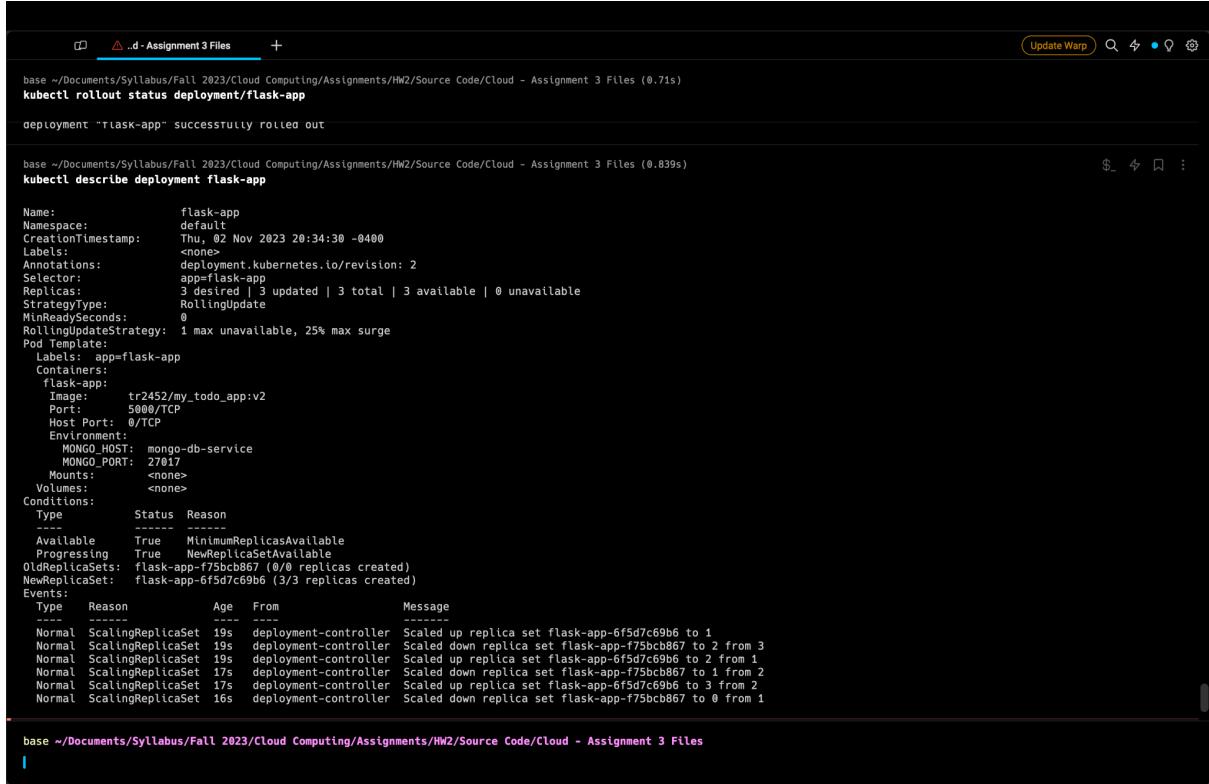
```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (1.626s)
kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
flask-app-f75bcbb87-2cs59  1/1    Running   0          40h
flask-app-f75bcbb87-9g944  1/1    Running   0          25h
flask-app-f75bcbb87-ttk59  1/1    Running   0          25h
flask-app-rc-2bm4q        1/1    Running   0          25h
flask-app-rc-95vjc        1/1    Running   0          25h
flask-app-rc-n6tsk        1/1    Running   0          25h
flask-app-rc-wz7p5        1/1    Running   0          25h
mongo-db-7485d8fdd6-6phxg 1/1    Running   0          40h
my-app-5b697fc7cd-xbqjp  1/1    Running   0          39h

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (1.296s)
kubectl apply -f Deployment_Rolling.yaml
deployment.apps/flask-app configured

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.71s)
kubectl rollout status deployment/flask-app
deployment "flask-app" successfully rolled out

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.839s)
kubectl describe deployment flask-app
Name:           flask-app
Namespace:      default
CreationTimestamp: Thu, 02 Nov 2023 20:34:30 -0400
Labels:          <none>
Annotations:    deployment.kubernetes.io/revision: 2
Selector:        app=flask-app
Replicas:       3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:   RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 1 max unavailable, 25% max surge
Pod Template:
  Labels:  app=flask-app
  Containers:
    flask-app:
      Image:   tr2452/my_todo_app:v2
      Port:    5000/TCP
      Host Port: 0/TCP
      Environment:
        MONGO_HOST: mongo_db_service
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files
```

-Checking the status of the rollout using 'kubectl rollout status'



```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.71s)
kubectl rollout status deployment/flask-app
deployment "flask-app" successfully rolled out

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.839s)
kubectl describe deployment flask-app
Name:           flask-app
Namespace:      default
CreationTimestamp: Thu, 02 Nov 2023 20:34:30 -0400
Labels:          <none>
Annotations:    deployment.kubernetes.io/revision: 2
Selector:        app=flask-app
Replicas:       3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:   RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 1 max unavailable, 25% max surge
Pod Template:
  Labels:  app=flask-app
  Containers:
    flask-app:
      Image:   tr2452/my_todo_app:v2
      Port:    5000/TCP
      Host Port: 0/TCP
      Environment:
        MONGO_HOST: mongo_db-service
        MONGO_PORT: 27017
      Mounts:   <none>
      Volumes:  <none>
  Conditions:
    Type     Status  Reason
    ----  -----  -----
    Available  True    MinimumReplicasAvailable
    Progressing  True    NewReplicaSetAvailable
OldReplicaSets: flask-app-f75bcbb87 (/0 replicas created)
NewReplicaSet:  flask-app-6f5d7c69b6 (3/3 replicas created)
Events:
  Type    Reason  Age   From            Message
  ----  -----  --  --  -----
  Normal  ScalingReplicaSet  19s  deployment-controller  Scaled up replica set flask-app-6f5d7c69b6 to 1
  Normal  ScalingReplicaSet  19s  deployment-controller  Scaled down replica set flask-app-f75bcbb867 to 2 from 3
  Normal  ScalingReplicaSet  19s  deployment-controller  Scaled up replica set flask-app-6f5d7c69b6 to 2 from 1
  Normal  ScalingReplicaSet  17s  deployment-controller  Scaled down replica set flask-app-f75bcbb867 to 1 from 2
  Normal  ScalingReplicaSet  17s  deployment-controller  Scaled up replica set flask-app-6f5d7c69b6 to 3 from 2
  Normal  ScalingReplicaSet  16s  deployment-controller  Scaled down replica set flask-app-f75bcbb867 to 0 from 1
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files
```

Test the Updated Application to Ensure That It Is Running With the New Version:

- Rolling update completed and Testing the new version of docker image and updated application for successful deployment.

The screenshot shows a Mac desktop environment. At the top, a Chrome browser window is open, displaying a 'ToDo Reminder' application. The browser's address bar shows the URL: `ac51a211671484079abbf4c59f2fc0e1-247441923.us-east-1.elb.amazonaws.com:5001/list`. The browser has several tabs open, including 'Best Method || C++ || JAVA', '(no subject) - tanmay71@gmail.com', and 'TODO with Flask'. Below the browser, a dark-themed 'ToDo Reminder' application window is visible. It features a header with 'ToDo Reminder', a navigation bar with 'ALL', 'Uncompleted', and 'Completed' buttons, and a search bar with 'Search Reference: Unique ID' and a 'Search' button. The main content area is titled 'To-DO LIST Version 2:' and contains a table with columns: Status, Task Name, Description Name, Date, Priority, Remove, and Modify. One task is listed: 'task1' (Status: X, Task Name: 'Name is', Description Name: ' ', Date: ' ', Priority: 'None', Remove: trash can icon, Modify: edit icon). Below the table, there is a form for adding a new task, labeled 'Add a Task', with fields for 'Taskname' (input field containing 'Taskname'), 'Enter Description here...' (text area), 'mm/dd/yyyy' (date input field), 'Priority' (dropdown menu), and a 'Create' button. At the very bottom of the screen is a dark dock bar containing icons for various Mac applications like Finder, Mail, Safari, and others. The desktop background is a solid light color.

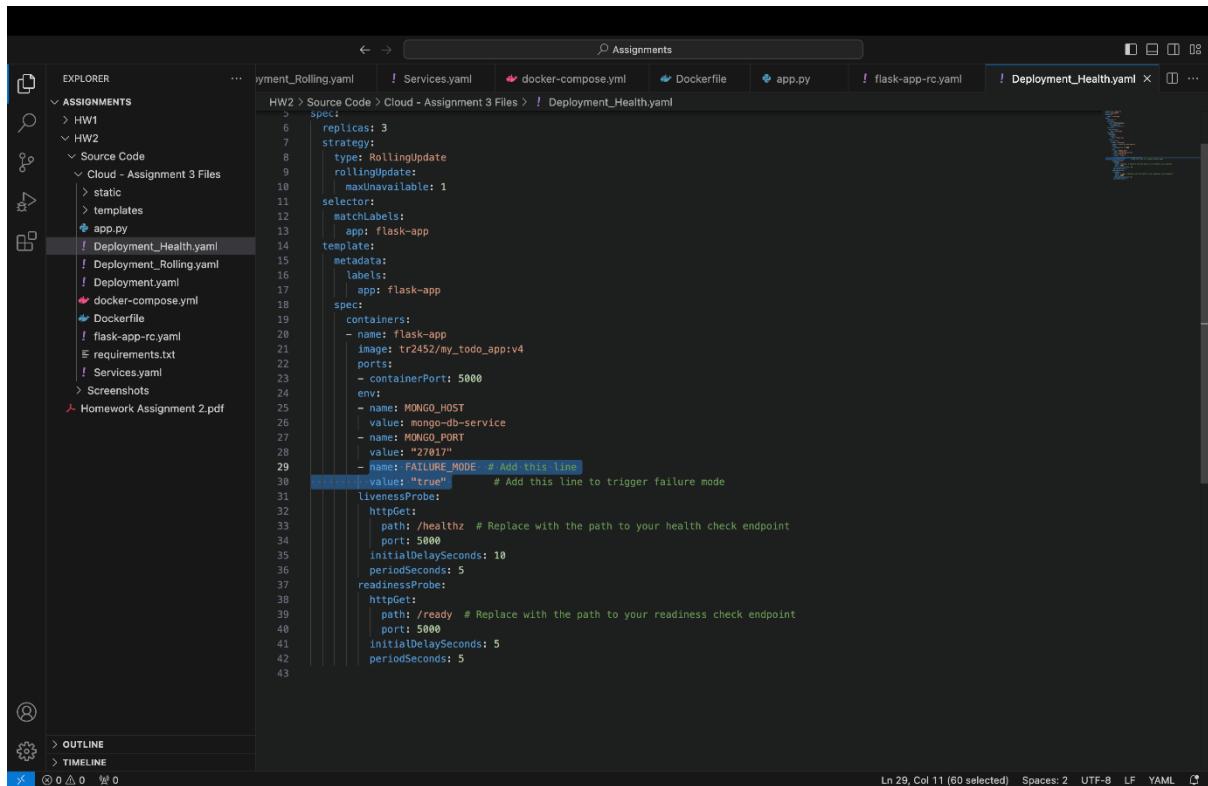
Part 7: Health Monitoring

Objective: Implement health checks to monitor application status and facilitate automatic recovery from failures.

Steps:

1. Configuring Probes:

- Added `livenessProbe` and `readinessProbe` to the deployment manifest.
- Configured probes to perform HTTP GET requests on specific endpoints at configured intervals.



```
! Deployment_Rolling.yaml   ! Services.yaml   docker-compose.yml   Dockerfile   app.py   flask-app-rc.yaml   Deployment_Health.yaml

Deployment_Health.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: flask-app
5  spec:
6    replicas: 3
7    strategy:
8      type: RollingUpdate
9      rollingUpdate:
10        maxUnavailable: 1
11    selector:
12      matchLabels:
13        app: flask-app
14    template:
15      metadata:
16        labels:
17          app: flask-app
18      spec:
19        containers:
20          - name: flask-app
21            image: tr2452/my_todo_app:v4
22            ports:
23              - containerPort: 5000
24            env:
25              - name: MONGO_HOST
26                value: mongo-db-service
27              - name: MONGO_PORT
28                value: "27017"
29              - name: FAILURE_MODE # Add this line
30                value: "true" # Add this line to trigger failure mode
31        livenessProbe:
32          httpGet:
33            path: /healthz # Replace with the path to your health check endpoint
34            port: 5000
35            initialDelaySeconds: 10
36            periodSeconds: 5
37        readinessProbe:
38          httpGet:
39            path: /ready # Replace with the path to your readiness check endpoint
40            port: 5000
41            initialDelaySeconds: 5
42            periodSeconds: 5
43
```

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows files like `HW2`, `Deployment_Health.yaml`, `Deployment_Rolling.yaml`, `Dockerfile`, `Dockerfile`, `flask-app-rc.yaml`, `app.py`, `Deployment_Health.yaml`, `Deployment_Rolling.yaml`, `Deployment.yaml`, `docker-compose.yml`, `requirements.txt`, `Services.yaml`, and `Screenshots`. A file named `Homework Assignment 2.pdf` is also listed.
- Current File:** The `app.py` file is open in the editor.
- Content of app.py:**

```
#!/usr/bin/env python3
# HW2 > Source Code > Cloud - Assignment 3 Files > app.py

# Import required libraries
from flask import Flask, render_template, request, jsonify
from pymongo import MongoClient
from bson.objectid import ObjectId
import json

# Initialize the application
app = Flask(__name__)

# Connect to MongoDB
client = MongoClient('mongodb://localhost:27017')
db = client['todolist']
todos = db['todos']

# Define routes and their corresponding functions
@app.route("/")
def index():
    todos_l = todos.find()
    if not todos_l:
        return render_template('index.html', a2=a2, todos=todos_l, t=title, h=heading, error="No such ObjectId is present")
    else:
        todos_l = todos.find()
        return render_template('searchlist.html', todos=todos_l, t=title, h=heading)

@app.route("/about")
def about():
    return render_template('credits.html', t=title, h=heading)

@app.route("/healthz")
def healthz():
    if failure_mode:
        # Simulate an application failure
        return "Service Unavailable", 503
    return "OK", 200

@app.route("/ready")
def ready():
    # Optionally check for readiness (e.g., database connections, etc.)
    return 'OK', 200

if __name__ == "__main__":
    env = os.environ.get("FLASK_ENV", 'development')
    port = int(os.environ.get('PORT', 5000))
    debug = False if env == 'production' else True
    app.run(host='0.0.0.0', port=port, debug=debug)
    #app.run(debug=True)
    #app.run(port=port, debug=debug)
    # Careful with the debug mode..
```

- Code Editor Tools:** Includes tabs for `assignments`, `mongdb_host`, and `flask-app-rc.yaml`. It also shows the current line (Ln 124), column (Col 1), and selected text (291).
- Bottom Status Bar:** Shows tab sizes (4), line feeds (LF), and Python as the language.

2. Added logic that will make the application/pod fail:

- Created a env variable in the deployment_health.yaml - that if set to true, will cause the application to fail.

```
- name: FAILURE_MODE # Add this line  
  value: "true"      # Add this line to trigger failure mode
```

The screenshot shows a code editor interface with multiple tabs and panels. The main area displays a Python file named `app.py` containing code for a Flask application. The code includes imports for Flask, pymongo, and bson.ObjectId, and defines routes for displaying tasks, marking them as completed, and listing completed tasks. The sidebar on the left shows a tree view of the project structure, including files like `HW2`, `Source Code`, `Cloud - Assignment 3 Files`, and `Deployment_Health.yaml`. The bottom status bar indicates the file is a Python script.

```
from flask import Flask, render_template, request, redirect, url_for # For flask implementation
from pymongo import MongoClient # Database connector
from bson.objectid import ObjectId # For ObjectId to work
from bson.errors import InvalidId # For catching InvalidId exception for ObjectId
import os

mongdb_host = os.environ.get('MONGO_HOST', 'localhost')
mongdb_port = int(os.environ.get('MONGO_PORT', '27017'))
failure_mode = os.environ.get('FAILURE_MODE', 'false').lower() == "true"

client = MongoClient(mongdb_host, mongdb_port) #Configure the connection to the database
db = client.camp2016 #select the database
todos = db.todos #select the collection

app = Flask(__name__)
title = "TODO with Flask"
heading = "Todo Reminder"
#modify ObjectId()

def redirect_url():
    return request.args.get('next') or \
           request.referrer or \
           url_for('index')

@app.route('/list')
def lists():
    #Display the all Tasks
    todos_l = todos.find()
    a1="active"
    return render_template('index.html',a1=a1,todos=todos_l,t=title,h=heading)

@app.route('/')
@app.route('/uncompleted')
def tasks():
    #Display the Uncompleted Tasks
    todos_l = todos.find({"done": "no"})
    a2="active"
    return render_template('index.html',a2=a2,todos=todos_l,t=title,h=heading)

@app.route('/completed')
def completed():
    #Display the Completed Tasks
```

3. Tested the changes locally and pushed the image v4 (With the error):

The screenshot shows a Mac desktop with a Chrome browser window open. The address bar shows 'localhost:5001/healthz'. The Network tab in the developer tools is selected, showing a single request named 'healthz' with a status of 503. The waterfall chart indicates a long duration, with most of the time spent between 20ms and 80ms. Below the Network tab, the Console tab shows the message 'Service Unavailable'.

The screenshot shows a Mac desktop with a Chrome browser window open. The address bar shows 'localhost:5001/ready'. The Network tab in the developer tools is selected, showing a single request named 'ready' with a status of 200. The waterfall chart shows a much shorter duration, mostly under 20ms. Below the Network tab, the Console tab shows the message 'OK'.

The changes work, so pushing the image.

 tr2452 / my_todo_app

Description

This repository does not have a description

 Last pushed: a few seconds ago

Tags

This repository contains 4 tag(s).

Tag	OS	Type	Pulled	Pushed
v4		Image	---	a few seconds ago
v3		Image	---	39 minutes ago
v2		Image	3 hours ago	3 hours ago
v1		Image	2 days ago	4 days ago

The Deployment-Health.yaml pulls the v4 image of the flask app.

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (1.189s)
kubectl apply -f Deployment_Health.yaml

deployment.apps/flask-app configured

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.756s)
kubectl get deployments

NAME      READY   UP-TO-DATE   AVAILABLE   AGE
flask-app  2/3     2           2           2d2h
mongo-db   1/1     1           1           2d2h
my-app     1/1     1           1           44h
```

4. Monitoring and Response:

- Monitored pod health using `kubectl describe pod`.
- Tested the liveness check by introducing a failure in the application and observed the automatic pod restart.

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.668s)
kubectl describe pod flask-app-b8d88cccd-drp5t

Host Port: @/TCP
State: Waiting
Reason: CrashLoopBackOff
Last State: Terminated
Reason: Completed
Exit Code: 0
Started: Sat, 04 Nov 2023 22:37:30 -0400
Finished: Sat, 04 Nov 2023 22:37:50 -0400
Ready: False
Restart Count: 3
Liveness: http-get http://:5000/healthz delay=10s timeout=1s period=5s #success=1 #failure=3
Readiness: http-get http://:5000/ready delay=5s timeout=1s period=5s #success=1 #failure=3
Environment:
  MONGO_HOST: mongo-db-service
  MONGO_PORT: 27017
  FAILURE_MODE: true
Mounts:
  /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-dp4gg (ro)
Conditions:
  Type      Status
  Initialized  True
  Ready      False
  ContainersReady  False
  PodScheduled  True
Volumes:
  kube-api-access-dp4gg:
    Type:       Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:  kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI:  true
QoS Class:  BestEffort
Node-Selectors: <none>
Tolerations:  node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type  Reason  Age From          Message
  Normal Scheduled  99s default-scheduler  Successfully assigned default/flask-app-b8d88cccd-drp5t to ip-172-31-90-87.ec2.internal
  Normal Pulling   98s kubelet        Pulling image "tr2452/my_todo_app:v4"
  Normal Pulled   97s kubelet        Successfully pulled image "tr2452/my_todo_app:v4" in 1.105s (1.105s including waiting)
  Normal Created   33s (x4 over 97s) kubelet        Created container flask-app
  Normal Started   33s (x4 over 97s) kubelet        Started container flask-app
  Warning Unhealthy 33s (x9 over 83s) kubelet  Liveness probe failed: HTTP probe failed with status code: 503
  Normal Killing   33s (x3 over 73s) kubelet        Container flask-app failed liveness probe, will be restarted
  Normal Pulled   33s (x3 over 73s) kubelet        Container image "tr2452/my_todo_app:v4" already present on machine

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files
```

The flask pods are failing due to the erroneous logic added with the error code specified by me.

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.643s)
kubectl get pods
mongo-db-748508fdd6-6phxq 2/2 Running 0 40s
my-app-5b697fc7cd-xbqjp 1/1 Running 0 44h

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.623s)
kubectl get pods
NAME READY STATUS RESTARTS AGE
flask-app-b8d88cccd-drp5t 0/1 CrashLoopBackOff 3 (28s ago) 114s
flask-app-b8d88cccd-m9plf 1/1 Running 0 4 (13s ago) 114s
flask-app-b8d88cccd-ttl9lq 1/1 CrashLoopBackOff 3 (18s ago) 103s
flask-app-rc-2bm4g 1/1 Running 0 30h
flask-app-rc-95vjc 1/1 Running 0 30h
flask-app-rc-n6tsk 1/1 Running 0 30h
flask-app-rc-wz7p5 1/1 Running 0 30h
mongo-db-748508fdd6-6phxq 1/1 Running 0 46h
my-app-5b697fc7cd-xbqjp 1/1 Running 0 44h

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.849s)
kubectl get pods
NAME READY STATUS RESTARTS AGE
flask-app-b8d88cccd-drp5t 1/1 Running 5 (9s ago) 2m30s
flask-app-b8d88cccd-m9plf 0/1 CrashLoopBackOff 4 (24s ago) 2m30s
flask-app-b8d88cccd-ttl9lq 1/1 Running 4 (54s ago) 2m19s
flask-app-rc-2bm4g 1/1 Running 0 30h
flask-app-rc-95vjc 1/1 Running 0 30h
flask-app-rc-n6tsk 1/1 Running 0 30h
flask-app-rc-wz7p5 1/1 Running 0 30h
mongo-db-748508fdd6-6phxq 1/1 Running 0 46h
my-app-5b697fc7cd-xbqjp 1/1 Running 0 44h

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.979s)
kubectl get pods
NAME READY STATUS RESTARTS AGE
flask-app-b8d88cccd-drp5t 0/1 CrashLoopBackOff 6 (14s ago) 4m40s
flask-app-b8d88cccd-m9plf 0/1 Running 6 (84s ago) 4m40s
flask-app-b8d88cccd-ttl9lq 1/1 Running 6 (104s ago) 4m29s
flask-app-rc-2bm4g 1/1 Running 0 30h
flask-app-rc-95vjc 1/1 Running 0 30h
flask-app-rc-n6tsk 1/1 Running 0 30h
flask-app-rc-wz7p5 1/1 Running 0 30h
mongo-db-748508fdd6-6phxq 1/1 Running 0 46h
my-app-5b697fc7cd-xbqjp 1/1 Running 0 44h

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files
```

The pods that are handled by deployment (Not by replication controller) are failing as the image pushed is causing it to do so.

But as they fail, the probe fails and so the system is trying to restart the pods.

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.774s)
kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
flask-app-6f5d7c69b6-gvs6w   1/1     Running   0          4h20m
flask-app-6f5d7c69b6-q46tr   1/1     Running   0          4h20m
flask-app-6f5d7c69b6-sgcr2   1/1     Running   0          4h20m
flask-app-rc-2bm4g          1/1     Running   0          29h
flask-app-rc-95vjc          1/1     Running   0          29h
flask-app-rc-n6tsk          1/1     Running   0          29h
flask-app-rc-w27p5          1/1     Running   0          29h
mongo-db-7485d8fdd6-6phxq   1/1     Running   0          45h
my-app-5b697fc7cd-xbqjp    1/1     Running   0          43h

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.706s)
kubectl get deployments
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
flask-app   3/3     3           3           2d1h
mongo-db   1/1     1           1           2d1h
my-app     1/1     1           1           43h
```

Part 8: Alerting

Objective: Establish an alerting mechanism to be notified of application issues.

Steps:

1. Set up Prometheus to Receive Alerts from Kubernetes:

- Deployed Prometheus using Helm, which included the Prometheus Operator, a set of custom resource definitions, and the Alertmanager.

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (10.91s)
brew install helm

Running `brew update --auto-update`...
==> Auto-updated Homebrew!
Updated 2 taps (homebrew/core and homebrew/cask).
==> New Formulae
appleaks      certsync      immer      modsurfer      presenterm      sigma-cli
appstream     cfricker     invoice      noir       python-docopt      tailspin
authz@       daktile       lager      orbuculum      python-mako      tlc
baob        der-ascii     libjuice      osctr-cli      raven       vet
bomber      dockerfile-language-server  libsail      phylum-cli      redwax-tool      zug
certgraph    gittuf       mediamtix      pmix       sclikit-image      music-decoy
deveco-studio          gstreamer-development      metamter      mitmproxy      orka3

You have 9 outdated formulae installed.

==> Downloading https://ghcr.io/v2/homebrew/core/helm/manifests/3.13.1
#####
==> Fetching helm
==> Downloading https://ghcr.io/v2/homebrew/core/helm/blobs/sha256:fe1b0ac507d63b7d127a934894e5c1d1050029ead01966c1575af5dffcd20b2
#####
==> Pouring helm--3.13.1.arm64_ventura.bottle.tar.gz
#####
Caveats
zsh completions have been installed to:
  /opt/homebrew/share/zsh/site-functions
  See here
  /opt/homebrew/Cellar/helm/3.13.1: 65 files, 53.7MB
==> Running `brew cleanup helm`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
```

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.489s)
helm version
version.BuildInfo{Version:"v3.13.1", GitCommit:"3547a4b5bf5edb5478ce352e18858d8a552a4110", GitTreeState:"clean", GoVersion:"go1.21.3"}

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (1.215s)
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
"prometheus-community" has been added to your repositories

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.782s)
helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "prometheus-community" chart repository
Update Complete. *Happy Helm-ing!*
```

- Installed Prometheus

- Deployed Prometheus within the Kubernetes cluster.

```

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (1.941s)
helm install prometheus prometheus-community/prometheus --namespace monitoring

NAME: prometheus
LAST DEPLOYED: Tue Nov 7 16:52:50 2023
NAMESPACE: monitoring
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
The Prometheus server can be accessed via port 80 on the following DNS name from within your cluster:
prometheus-server.monitoring.svc.cluster.local

Get the Prometheus server URL by running these commands in the same shell:
export POD_NAME=$(kubectl get pods --namespace monitoring -l "app.kubernetes.io/name=prometheus,app.kubernetes.io/instance=prometheus" -o jsonpath=".items[0].metadata.name")
kubectl --namespace monitoring port-forward $POD_NAME 9090

The Prometheus alertmanager can be accessed via port 9093 on the following DNS name from within your cluster:
prometheus-alertmanager.monitoring.svc.cluster.local

Get the AlertManager URL by running these commands in the same shell:
export POD_NAME=$(kubectl get pods --namespace monitoring -l "app.kubernetes.io/name=alertmanager,app.kubernetes.io/instance=prometheus" -o jsonpath=".items[0].metadata.name")
kubectl --namespace monitoring port-forward $POD_NAME 9093

#####
##### WARNING: Pod Security Policy has been disabled by default since #####
##### it deprecated after K8s 1.25+. use #####
##### (index .Values "prometheus-node-exporter" "rbac" #####
##### . "pspEnabled") with (index .Values #####
##### "prometheus-node-exporter" "rbac" "pspAnnotations") #####
##### in case you still need it. #####
#####

The Prometheus PushGateway can be accessed via port 9091 on the following DNS name from within your cluster:
prometheus-prometheus-pushgateway.monitoring.svc.cluster.local

Get the PushGateway URL by running these commands in the same shell:
export POD_NAME=$(kubectl get pods --namespace monitoring -l "app=prometheus-pushgateway,component=pushgateway" -o jsonpath=".items[0].metadata.name")
kubectl --namespace monitoring port-forward $POD_NAME 9091

For more information on running Prometheus, visit:
https://prometheus.io/
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
alertmanager-operated	ClusterIP	None	<none>	9093/TCP,9094/TCP,9094/UDP	45m
flask-mongo-service	LoadBalancer	10.100.55.13	ac51a211671484079abbf4c59f2fc@e1-247441923.us-east-1.elb.amazonaws.com	5001:31438/TCP	4d
kubernetes	ClusterIP	10.100.0.1	<none>	443/TCP	4d
mongo-db-service	ClusterIP	10.100.51.73	<none>	27017/TCP	4d
my-pro-grafana	ClusterIP	10.100.88.250	<none>	80/TCP	45m
my-pro-kube-prometheus-sta-alertmanager	ClusterIP	10.100.146.159	<none>	9093/TCP,8080/TCP	45m
my-pro-kube-prometheus-sta-operator	ClusterIP	10.100.25.89	<none>	443/TCP	45m
my-pro-kube-prometheus-sta-prometheus	ClusterIP	10.100.101.182	<none>	9090/TCP,8080/TCP	45m
my-pro-kube-state-metrics	ClusterIP	10.100.88.24	<none>	8080/TCP	45m
my-pro-prometheus-node-exporter	ClusterIP	10.100.214.152	<none>	9100/TCP	45m
prometheus-operated	ClusterIP	None	<none>	9090/TCP	45m

2. Configured Prometheus to Generate Alerts:

- Created a PrometheusRule resource with rules to define when alerts should be triggered, such as when a pod restarts more than a specified number of times.

```

apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"monitoring.coreos.com/v1","kind":"PrometheusRule","metadata":{"annotations":{"meta.helm.sh/release-name":"my-pro","meta.helm.sh/release-namespace":"default","prometheus-operator-validate":true}},"creationTimestamp":"2023-11-06T01:03:40Z","generation":3,"labels":{},"spec":{"groups":[{"name": "example","rules": [{"alert": "PodCrashLooping","annotations": {"description": "Pod {{ $labels.namespace }}/{{ $labels.pod }} is crash looping."}, "summary": "Pod is crash looping", "expr": "kube_pod_container_status_restarts_total{job=\"kube-state-metrics\"} > 1", "for": "1m", "labels": {"severity": "critical"}]}]}}}
  creationTimestamp: "2023-11-06T01:03:40Z"
  generation: 3
  labels:
    app: kube-prometheus-stack
    app.kubernetes.io/instance: my-pro
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/part-of: kube-prometheus-stack
    app.kubernetes.io/version: 52.1.0
    chart: kube-prometheus-stack-52.1.0
    heritage: Helm
    release: my-pro
    name: my-pro-kube-prometheus-sta-general.rules
    namespace: default
    resourceVersion: "1457469"
    uid: 9894e64d-3c15-4c2a-aede-5f8c0ae13238
  spec:
    groups:
    - name: example
      rules:
      - alert: PodCrashLooping
        annotations:
          description: Pod {{ $labels.namespace }}/{{ $labels.pod }} is crash looping.
          summary: Pod is crash looping
          expr: kube_pod_container_status_restarts_total{job="kube-state-metrics"} > 1
          for: 1m
          labels:
            severity: critical

```

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.738s)
kubectl get prometheusrules my-pro-kube-prometheus-sta-general.rules -n default -o yaml > my-rule-file.yaml

base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (1.137s)
kubectl apply -f my-rule-file.yaml
prometheusrule.monitoring.coreos.com/my-pro-kube-prometheus-sta-general.rules configured
```

3. Set Up Notification Service to Receive Alerts from Prometheus:

- Configured Alertmanager to forward alerts to a Slack channel using a webhook URL provided by Slack.

```
global:
  resolve_timeout: 5m
  lombard_rules:
  - equal:
    - namespace
    - alertname
    source_matchers:
      - severity = critical
    target_matchers:
      - severity == warning|info
    - equal:
      - namespace
      - alertname
    source_matchers:
      - severity = warning
    target_matchers:
      - severity = info
    - equal:
      - namespace
      - source_matchers:
        - alertname = InfoInhibitor
      target_matchers:
        - severity = info
    receivers:
      - name: 'slack-notifications'
      slack_configs:
      - endpoint: 'https://hooks.slack.com/services/T064NBS2DPX/B064TNH#4C/7AUJV91B69yOYH98Qc0hm6b'
        send_resolved: true
        text: |
          {{range .Alerts}}
            *{{.Annotations.summary}} - `{{.Labels.severity}}`*
            {{.Annotations.description}}
            {{.Spec.group}}
            {{range .Labels.SortedPairs}} *{{.Name}}*: `{{.Value}}`{{end}}
          {{end}}
        {{end}}
    - name: 'null'
  routes:
  - receiver: 'null'
  group_by: ['namespace']
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 12h
  routes:
  - match:
    - alertname: PodCrashLooping
    receiver: 'slack-notifications'
  templates:
  - /etc/alertmanager/config/*.tmpl
```

4. Test the Alerting System

- Deployed a pod designed to fail, causing it to enter a crash loop, which would trigger the PodCrashLooping alert.

```
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: test-crash-loop-pod
5    namespace: default
6  spec:
7    containers:
8      - name: crash-loop-container
9        image: busybox
10       command: ["false"]
```

```
base ~/Documents/Syllabus/Fall 2023/Cloud Computing/Assignments/HW2/Source Code/Cloud - Assignment 3 Files (0.854s)
kubectl get pods
NAME                                READY   STATUS        RESTARTS   AGE
alertmanager-my-pro-kube-prometheus-sta-alertmanager-0   2/2    Running      0          21m
failing-pod                           0/1    CrashLoopBackOff  1047 (29s ago)  3d16h
flask-app-6ffd7c69b6-ctkdq           1/1    Running      0          4d10h
flask-app-6f5d7c69b6-qd6cp           1/1    Running      0          4d12h
flask-app-6f5d7c69b6-v75mw           1/1    Running      0          3d20h
flask-app-rc-2bm4g                  1/1    Running      0          5d22h
flask-app-rc-95vjc                 1/1    Running      0          5d22h
flask-app-rc-n6tsk                 1/1    Running      0          5d22h
flask-app-rc-t4g2w                  1/1    Running      0          3d21h
flask-app-rc-w6ng9                  1/1    Running      0          4d12h
flask-app-rc-zwzp9                  1/1    Running      0          4d12h
mongo-db-7485d8fd6-6phxq           1/1    Running      0          6d14h
my-app-5b697fc7cd-dngjc            1/1    Running      0          3d20h
my-pro-grafana-65fc586568-5cd6n     3/3    Running      0          3d17h
my-pro-kube-prometheus-sta-operator-6bb7bccfb7-z788q       1/1    Running      0          3d17h
my-pro-kube-state-metrics-85997cfbc4-lvk7f                 1/1    Running      0          3d17h
my-pro-prometheus-node-exporter-kxmls             1/1    Running      0          3d17h
my-pro-prometheus-node-exporter-vrgq7             1/1    Running      0          3d17h
prometheus-my-pro-kube-prometheus-sta-prometheus-0       2/2    Running      0          3d17h
test-crash-loop-pod                  0/1    CrashLoopBackOff  7 (4m55s ago)  15m
```

5. Checked the designated Slack channel for the alert notification.

Prometheus Alerts APP 13:02

[FIRING:1] default (PodCrashLooping failing-container http 172.31.76.136:8080 kube-state-metrics failing-pod default/my-pro-kube-prometheus-sta-prometheus my-pro-kube-state-metrics critical 793dc94-c424-43c1-8627-796793642496)

Alert: Pod is crash looping - **critical**

Description: Pod default/failing-pod is crash looping.

Details:

- alertname: **PodCrashLooping**
- container: **failing-container**

Show more

Prometheus Alerts APP 13:22

[FIRING:2] default (PodCrashLooping http 172.31.76.136:8080 kube-state-metrics default/my-pro-kube-prometheus-sta-prometheus my-pro-kube-state-metrics critical)

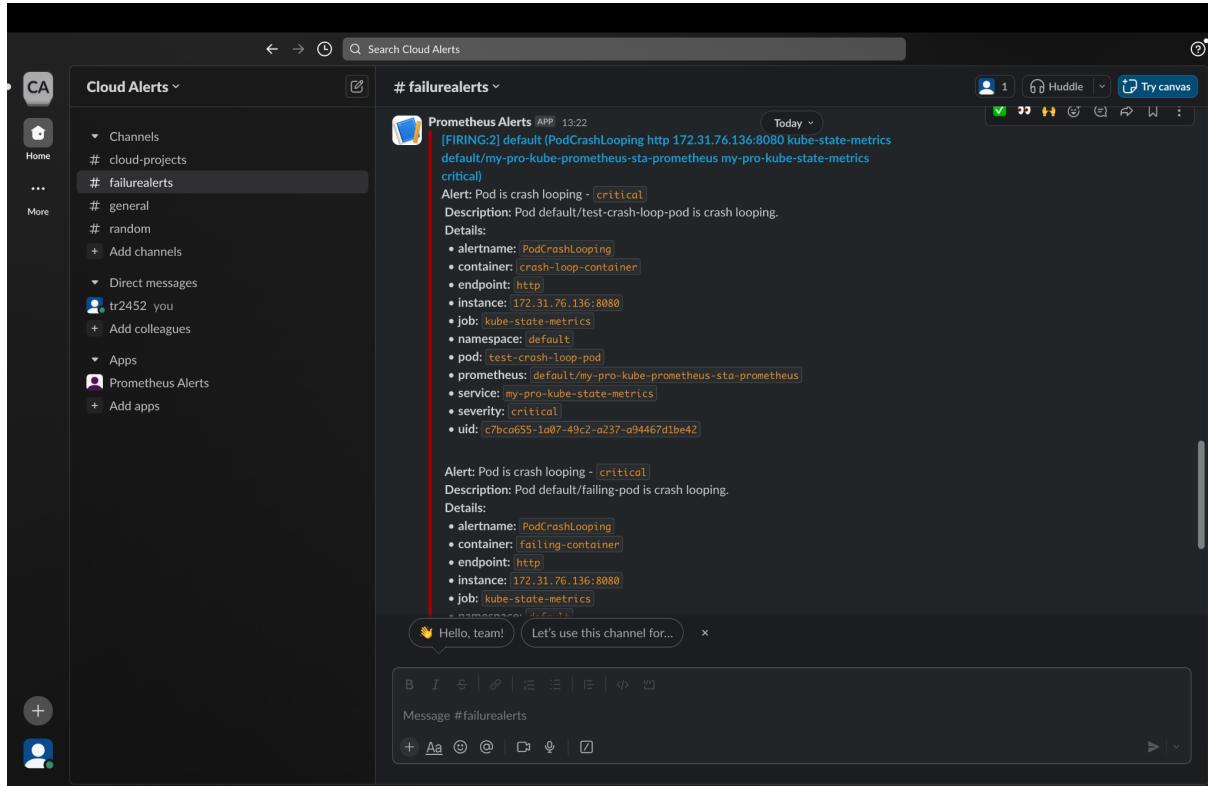
Alert: Pod is crash looping - **critical**

Description: Pod default/test-crash-loop-pod is crash looping.

Details:

- alertname: **PodCrashLooping**
- container: **crash-loop-container**

Show more



This represents the successful generation of alerts and therefore the completion of the assignment.