

Assignment 2 - CS-GY 9223 Cloud Computing

Submitted by Pavithra Rajan (pr2725)

Task 1 and 2: Containerizing the Application on Docker

- Create **Dockerfile** and **docker-compose.yaml**

Docker compose commands

```
docker compose build
docker compose up -d
docker compose logs mongodb
docker compose logs web
```

Pushing image to DockerHub

The image shows a terminal window and a Docker Hub repository page. The terminal window displays the command `docker push pavithra22rajan/todo-app:v1` and its output, which includes the push of the image to Docker Hub. The Docker Hub page shows the repository `pavithra22rajan/todo-app` with tags `v1`, `v3`, and `v4`. The page also displays the Docker commands for pushing and pulling the image.

```
pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS [main] docker push pavithra22rajan/todo-app:v1
The push refers to repository [docker.io/pavithra22rajan/todo-app]
1ceda713dfa5: Pushed
bc762e0ebf77: Pushed
d3ffdfbacc07: Pushed
c8f6b54339a8: Mounted from library/python
298992e09a03: Mounted from library/python
4f237755fbae: Mounted from library/python
d7c97cb6f1fe: Mounted from library/python
v1: digest: sha256:9300cd1be20566e0ea61a278f204bb625e6e6cc75c91f0bd83e1deaea277f919 size: 1787
```

Docker Hub repository page for `pavithra22rajan/todo-app`. The page shows the repository details, including the last pushed time (3 hours ago) and the repository size (263.8 MB). The tags section lists the following tags:

TAG	Digest	OS/ARCH	Last pull	Compressed size
v1	9300cd1be205	linux/amd64	less than 1 day	54.73 MB
v3	d65965f6440e	linux/amd64	less than 1 day	56.73 MB
v4				

Task 3: Deploying the Application on Minikube

```
# Start a minikube cluster
minikube start

# Create a PVC
```

```

kubectl apply -f mongo-pvc.yaml

# Create a pod to store the data
kubectl apply -f mongo-deployment.yaml

# Internal service for MongoDB as ClusterIP
kubectl apply -f mongo-service.yaml

# deploy the Flask application using the image pushed above with same name
kubectl apply -f flask-deployment.yaml

# create a Node port for external access
kubectl apply -f flask-service.yaml

# View all pods
kubectl get all

# To get the service URL to access
minikube service flask-todo-service --url

# Search on selector type
kubectl get pods -l tier=frontend

# to know what port type and IP
kubectl get svc flask-todo-service

# logs per pod
kubectl logs -f flask-todo-deployment-b7fb88598-fksqb

```

Service details

```

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS [main] kubectl get all
NAME                                READY    STATUS    RESTARTS   AGE
pod/flask-todo-deployment-b7fb88598-9hwfk  1/1      Running   1 (14h ago)  24h
pod/flask-todo-deployment-b7fb88598-fksqb  1/1      Running   1 (14h ago)  24h
pod/mongodb-deployment-685ff9fd7f-js2ks   1/1      Running   1 (14h ago)  25h

NAME                                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
service/flask-todo-service           NodePort    10.99.104.251  <none>         80:30740/TCP     24h
service/kubernetes                   ClusterIP   10.96.0.1      <none>         443/TCP          5d19h
service/mongodb-service              ClusterIP   10.103.193.104 <none>         27017/TCP        24h

NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/flask-todo-deployment  2/2      2              2            24h
deployment.apps/mongodb-deployment    1/1      1              1            25h

NAME                                DESIRED    CURRENT    READY    AGE
replicaset.apps/flask-todo-deployment-b7fb88598  2          2          2        24h
replicaset.apps/mongodb-deployment-685ff9fd7f    1          1          1        25h

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS [main] minikube service flask-todo-service --url
http://192.168.49.2:30740

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS [main] kubectl get pods -l tier=frontend
NAME                                READY    STATUS    RESTARTS   AGE
flask-todo-deployment-b7fb88598-9hwfk  1/1      Running   1 (14h ago)  24h
flask-todo-deployment-b7fb88598-fksqb  1/1      Running   1 (14h ago)  24h

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS [main] kubectl get svc flask-todo-service
NAME                                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
flask-todo-service                   NodePort    10.99.104.251  <none>         80:30740/TCP     24h

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS [main] kubectl get pods
NAME                                READY    STATUS    RESTARTS   AGE
flask-todo-deployment-b7fb88598-9hwfk  1/1      Running   1 (14h ago)  24h
flask-todo-deployment-b7fb88598-fksqb  1/1      Running   1 (14h ago)  24h
mongodb-deployment-685ff9fd7f-js2ks   1/1      Running   1 (14h ago)  25h

```

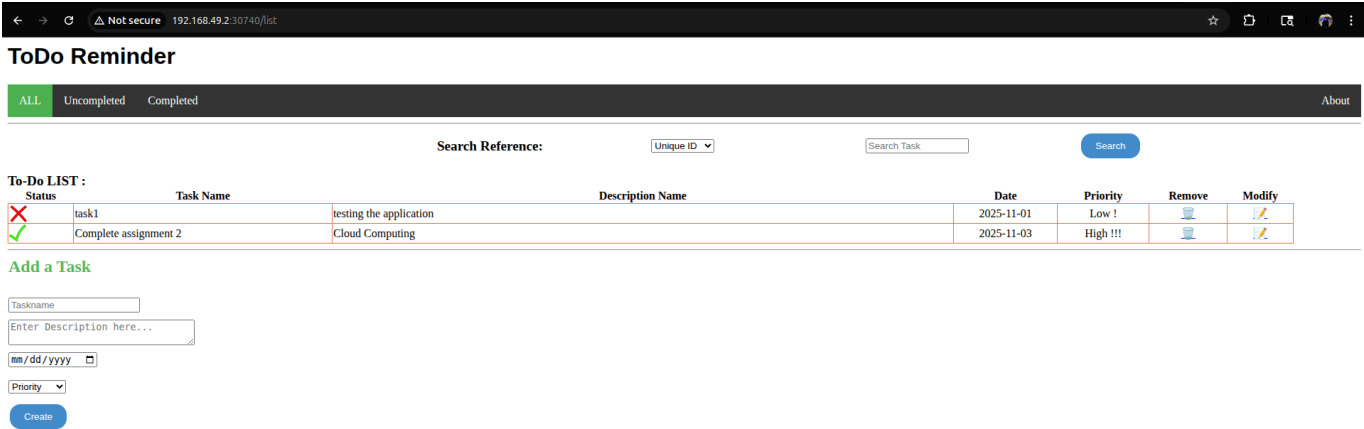
Frontend replica details

```
pavithra@shortcake ~/. ./CS-GY-9223-Cloud-Assg2-EKS main kubectll get pods -l tier=frontend
NAME                                READY  STATUS   RESTARTS  AGE
flask-todo-deployment-b7fb88598-9hwwk 1/1    Running  0         3m15s
flask-todo-deployment-b7fb88598-fksqb 1/1    Running  0         3m15s
```

Deleting a pod

```
pavithra@shortcake ~/. ./CS-GY-9223-Cloud-Assg2-EKS main kubectll get pods
NAME                                READY  STATUS   RESTARTS  AGE
flask-todo-deployment-b7fb88598-9hwwk 1/1    Running  1 (14h ago)  24h
flask-todo-deployment-b7fb88598-fksqb 1/1    Running  1 (14h ago)  24h
mongodb-deployment-685ff9fd7f-js2ks  1/1    Running  1 (14h ago)  25h
pavithra@shortcake ~/. ./CS-GY-9223-Cloud-Assg2-EKS main kubectll delete pod flask-todo-deployment-b7fb88598-9hwwk
pod "flask-todo-deployment-b7fb88598-9hwwk" deleted from default namespace
pavithra@shortcake ~/. ./CS-GY-9223-Cloud-Assg2-EKS main kubectll get pods
NAME                                READY  STATUS   RESTARTS  AGE
flask-todo-deployment-b7fb88598-fksqb 1/1    Running  1 (14h ago)  24h
flask-todo-deployment-b7fb88598-lfcfl 1/1    Running  0         4s
mongodb-deployment-685ff9fd7f-js2ks  1/1    Running  1 (14h ago)  25h
pavithra@shortcake ~/. ./CS-GY-9223-Cloud-Assg2-EKS main
```

Running local instance









Task 4: Deploying the Application on AWS EKS

- Create a custom EKS cluster in AWS
 - **create Cluster IAM role:** This is the most critical IAM role for the EKS cluster itself. The EKS control plane (the Kubernetes master components) needs permission to call AWS APIs on your behalf, primarily to manage resources like the EC2 instances that serve as the worker nodes and to create ELBs. When you deploy an LB, the EKS cluster uses its IAM role to ask AWS to provision the new ELB. Without this role, the service will get stuck in a "Pending" state.

<input type="checkbox"/>		AmazonEKSBloclStoragePolicy	AWS managed	1
<input type="checkbox"/>		AmazonEKSClusterPolicy	AWS managed	1
<input type="checkbox"/>		AmazonEKSComputePolicy	AWS managed	1
<input type="checkbox"/>		AmazonEKSLoadBalancingPolicy	AWS managed	1
<input type="checkbox"/>		AmazonEKSNetworkingPolicy	AWS managed	1

- **create EKS Node Group IAM Role:** The worker nodes (EC2 instances) also need an IAM role. This role grants the worker nodes permissions to join the EKS cluster, pull Docker images from AWS ECR (Elastic Container Registry), send logs and metrics to AWS services like CloudWatch.

<input type="checkbox"/>	Policy name 	 Type	 Attached entities
<input type="checkbox"/>	 AmazonEC2ContainerRegistryReadOnly	AWS managed	1
<input type="checkbox"/>	 AmazonEKS_CNI_Policy	AWS managed	1
<input type="checkbox"/>	 AmazonEKSWorkerNodePolicy	AWS managed	1

- Enable Prometheus and CloudWatch
- Create Pod Identity IAM role for service account

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "pods.eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

EKSConfig

```
aws eks update-kubeconfig --region us-east-1 --name todo-app-v1
```

Updated the above user with new inline policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "eks:DescribeCluster",
        "eks:ListClusters",
        "eks:AccessKubernetesApi",
        "eks:ListUpdates",
        "eks:ListNodegroups",
        "eks:DescribeNodegroup"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "*"
    }
  ]
}
```

```

        "Condition": {
          "StringEquals": {
            "iam:PassedToService": "eks.amazonaws.com"
          }
        }
      }
    ]
  }
}

```

- Set the service spec type as **LoadBalancer**.

```
kubectl patch svc flask-todo-service -p '{"spec": {"type": "LoadBalancer"}}'
```

```

pavithra@shortcake ~/.../CS-GY-9223-Cloud-Assg2-EKS main kubectl describe svc flask-todo-service
Name: flask-todo-service
Namespace: default
Labels: app=todo-app, tier=frontend
Annotations: updated-trigger: 1762051501
Selector: app=todo-app, tier=frontend
Type: LoadBalancer
IP Family Policy: SingleStack
IP Families: IPv4
IP: 10.100.3.218
IPs: 10.100.3.218
LoadBalancer Ingress: a8125644190f44860bfc3d2730d4b515-1462299559.us-east-1.elb.amazonaws.com
Port: <unset> 80/TCP
TargetPort: 5000/TCP
NodePort: <unset> 32693/TCP
Endpoints: 172.31.2.112:5000,172.31.2.113:5000
Session Affinity: None
External Traffic Policy: Cluster
Internal Traffic Policy: Cluster
Events:
  Type     Reason              Age   From              Message
  ----     -
  Normal   EnsuringLoadBalancer 18s   service-controller Ensuring load balancer
  Normal   EnsuredLoadBalancer 14s   service-controller Ensured load balancer

```

- Modified IAM policy of user to have **iam*** and **cloudformation*** owing to authorization issues.

Service account creation

```

eksctl utils associate-iam-oidc-provider --region=us-east-1 --cluster=todo-app-v1 --approve
2025-11-01 23:49:16 [i] will create IAM Open ID Connect provider for cluster "todo-app-v1" in "us-east-1"
2025-11-01 23:49:16 [✓] created IAM Open ID Connect provider for cluster "todo-app-v1" in "us-east-1"

```

```

eksctl create iamserviceaccount \
  --cluster=todo-app-v1 \
  --namespace=kube-system \

```

```
--name=ebs-csi-controller-sa \  
--attach-policy-arn=arn:aws:iam::aws:policy/service-  
role/AmazonEBSCSIDriverPolicy \  
--override-existing-serviceaccounts \  
--approve
```

Storage Class for PVC

- created a storage class `gp2-eks` and added that to `mongo-pvc`
- applied the storage class followed by the pvc

```
kubectl get storageclass
```

NAME	PROVISIONER	RECLAIMPOLICY	VOLUMEBINDINGMODE
gp2	kubernetes.io/aws-ebs	Delete	WaitForFirstConsumer
ALLOWVOLUMEEXPANSION	AGE		
false	5h29m		
gp2-eks	ebs.csi.eks.amazonaws.com	Delete	WaitForFirstConsumer
false	2m58s		

```
kubectl apply -f gp2-eks-sc.yaml
```

Task 5: Deployments and ReplicaSets

- Specified the replicas in `spec.replica` in `flask-deployment.yaml`.
- Deploy the changes and check the replica details with `kubectl get rs`. We can see the DESIRED, CURRENT and READY replicas.
- scale-up to have 5 replicas via `kubectl`.

- scale-down back to 2 replicas.

```

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS P main$ kubectl get deploy flask-todo-deployment
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
flask-todo-deployment 2/2     2            2           3h5m

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS P main$ kubectl get rs
NAME                DESIRED   CURRENT   READY   AGE
flask-todo-deployment-646dc6d6d7 2          2         2       3h6m
mongodb-deployment-7c4d97b4fb 1          1         1       41m

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS P main$ kubectl get pods -l tier=frontend
NAME                READY   STATUS    RESTARTS   AGE
flask-todo-deployment-646dc6d6d7-2fw6z 1/1     Running   0          3h6m
flask-todo-deployment-646dc6d6d7-fv2s6 1/1     Running   0          3h6m

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS P main$ kubectl delete pod flask-todo-deployment-646dc6d6d7-2fw6z
pod "flask-todo-deployment-646dc6d6d7-2fw6z" deleted from default namespace

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS P main$ kubectl get pods -l tier=frontend
NAME                READY   STATUS    RESTARTS   AGE
flask-todo-deployment-646dc6d6d7-fv2s6 1/1     Running   0          3h6m
flask-todo-deployment-646dc6d6d7-vbwvd 1/1     Running   0          5s

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS P main$ kubectl scale deployment flask-todo-deployment --replicas=5
deployment.apps/flask-todo-deployment scaled

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS P main$ kubectl get pods -l tier=frontend
NAME                READY   STATUS    RESTARTS   AGE
flask-todo-deployment-646dc6d6d7-c6g76 1/1     Running   0          4s
flask-todo-deployment-646dc6d6d7-fv2s6 1/1     Running   0          3h8m
flask-todo-deployment-646dc6d6d7-n6bch 1/1     Running   0          4s
flask-todo-deployment-646dc6d6d7-rmfjc 1/1     Running   0          4s
flask-todo-deployment-646dc6d6d7-vbwvd 1/1     Running   0          101s

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS P main$ kubectl scale deployment flask-todo-deployment --replicas=2
deployment.apps/flask-todo-deployment scaled

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS P main$ kubectl get pods -l tier=frontend
NAME                READY   STATUS    RESTARTS   AGE
flask-todo-deployment-646dc6d6d7-fv2s6 1/1     Running   0          3h8m
flask-todo-deployment-646dc6d6d7-vbwvd 1/1     Running   0          112s

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS P main$ kubectl get rs
NAME                DESIRED   CURRENT   READY   AGE
flask-todo-deployment-646dc6d6d7 2          2         2       3h11m
mongodb-deployment-7c4d97b4fb 1          1         1       47m

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS P main$ kubectl scale deployment flask-todo-deployment --replicas=5
deployment.apps/flask-todo-deployment scaled

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS P main$ kubectl get rs
NAME                DESIRED   CURRENT   READY   AGE
flask-todo-deployment-646dc6d6d7 5          5         5       3h11m
mongodb-deployment-7c4d97b4fb 1          1         1       47m

```

Task 6: Rolling update strategy

- Made a minor change to the `app.py` to build a new image. This was by adding a space in the header of the application.
- Build the image and push it.

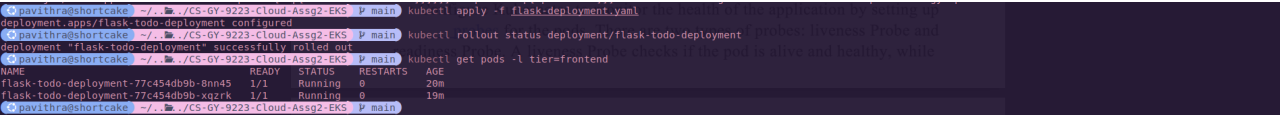
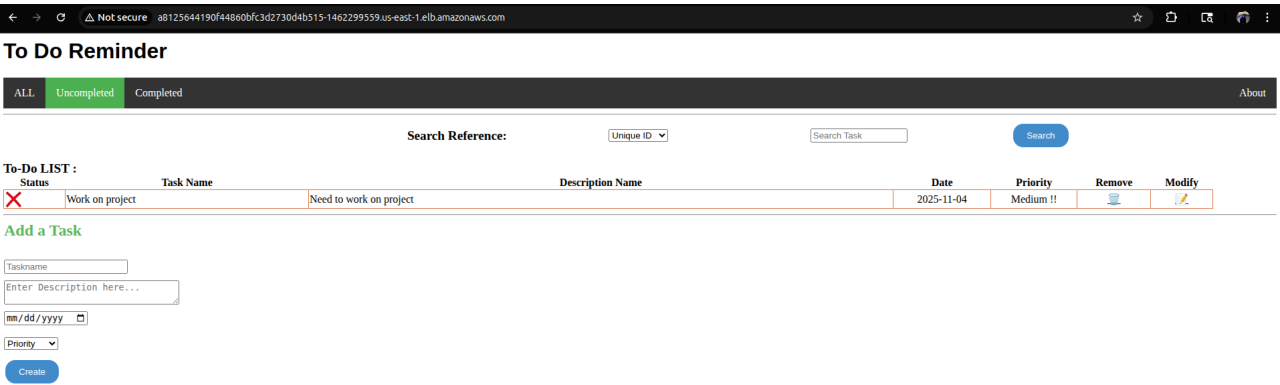
```

pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS P main$ docker build -t pavithra22rajan/todo-app:v2 .
[+] Building 9.8s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/python:3.9-slim
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> [internal] transferring context: 2B
=> [1/4] FROM docker.io/library/python:3.9-slimsha256:2d9776910b16bd338d3060f261f53f144965f755599aabbacdale13cf1731b1b
=> [internal] load build context
=> [internal] transferring context: 1.12MB
=> [2/4] WORKDIR /app
=> [3/4] COPY . /app
=> [4/4] RUN pip install --no-cache-dir -r requirements.txt
=> exporting image
=> writing image sha256:bb8145c06cdf6d2ebe9d36e82ec35cb4ea06263a8cea336cca9cd37b6572714
=> naming to docker.io/pavithra22rajan/todo-app:v2
The push refers to repository [docker.io/pavithra22rajan/todo-app]
23400c4511ae: Pushed
e93a1e7a7a3f: Pushed
d3fddfbacc07: Layer already exists
13f6d54539a8: Layer already exists
298992e09a03: Layer already exists
4f237755fb0e: Layer already exists
07c97cb6f1fe: Layer already exists
v2: digest: sha256:811a2d6dc2fe8dd404210a9aa147606ae22647b5380c3f60ee30f7f2afdc2b size: 1787

```

- Make changes to `flask-deployment.yaml` to have the right tag.

- The status can be monitored with `kubectl`.

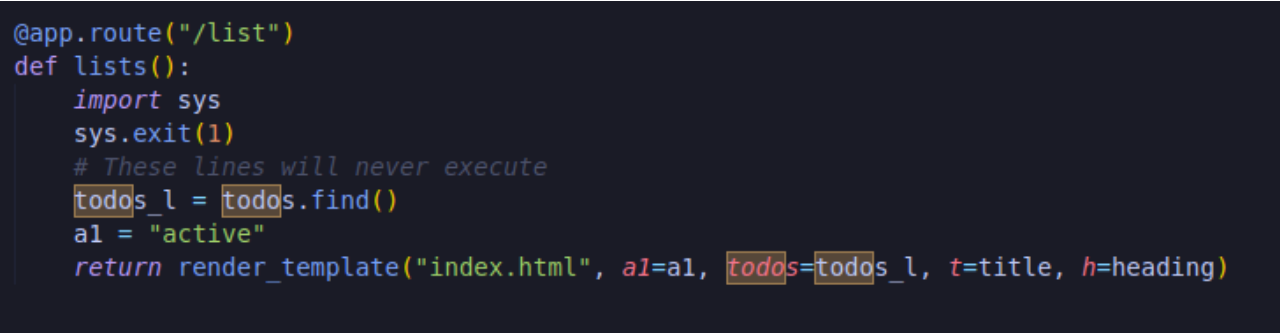
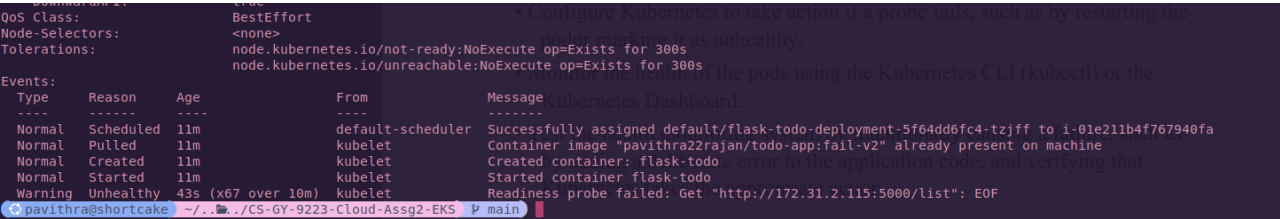


Task 7: Health monitoring

- Added liveness and readiness probe to `flask-deployment.yaml`.

Readiness probe

- Added a `sys.exit` in `/list` endpoint and re-applied the changes.



Liveness probe

- Similarly, did the above for / root endpoint. Here, we can see that the pod keeps restarting.

```

tolerations:
  - key: "kubernetes.io/not-ready"
    operator: "Exists"
    value: "300s"
  - key: "kubernetes.io/unreachable"
    operator: "Exists"
    value: "300s"

Events:
Type      Reason      Age           From          Message
----      -
Normal    Scheduled   106s         default-scheduler   Successfully assigned default/flask-todo-deployment-68d595cbfc-8qhl5 to i-01e211b4f767940fa
Normal    Pulling     106s         kubelet          Pulling image "pavithra22rajan/todo-app:fail-v3"
Normal    Pulled      105s         kubelet          Successfully pulled image "pavithra22rajan/todo-app:fail-v3" in 1.127s (1.127s including waiting). Image size: 59874943 bytes.
Normal    Killing     27s (x2 over 67s) kubelet          Container flask-todo failed liveness probe, will be restarted
Normal    Created     26s (x3 over 195s) kubelet          Created container: flask-todo
Normal    Started     26s (x3 over 195s) kubelet          Started container flask-todo
Normal    Pulled      26s (x2 over 66s) kubelet          Container image "pavithra22rajan/todo-app:fail-v3" already present on machine
Warning   Unhealthy   7s (x7 over 87s) kubelet          Liveness probe failed: Get "http://172.31.2.113:5000/": EOF
pavithra@shortcake: ~/.../CS-GY-9223-Cloud-Assg2-EKS $ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
flask-todo-deployment-68d595cbfc-8qhl5 1/1     Running   2 (30s ago) 11s
flask-todo-deployment-68d595cbfc-px45f 1/1     Running   2 (15s ago) 9s
mongodb-deployment-7c4d97b4fb-wb74c    1/1     Running   0           3h11m

```

Task 8: Alerting [Extra Credit]

```
kubectl create secret generic prometheus-kube-prometheus-alertmanager -n monitoring --from-file=alertmanager.yaml=alertmanager.yaml
```

```

helm upgrade --install prometheus prometheus-community/kube-prometheus-stack --namespace monitoring \
  --set alertmanager.enabled=true \
  --set alertmanager.alertmanagerSpec.routePrefix=/ \
  --set alertmanager.ingress.enabled=false \
  --set
prometheus.prometheusSpec.serviceMonitorSelectorNilUsesHelmValues=false \
  --set
prometheus.prometheusSpec.podMonitorSelectorNilUsesHelmValues=false

```

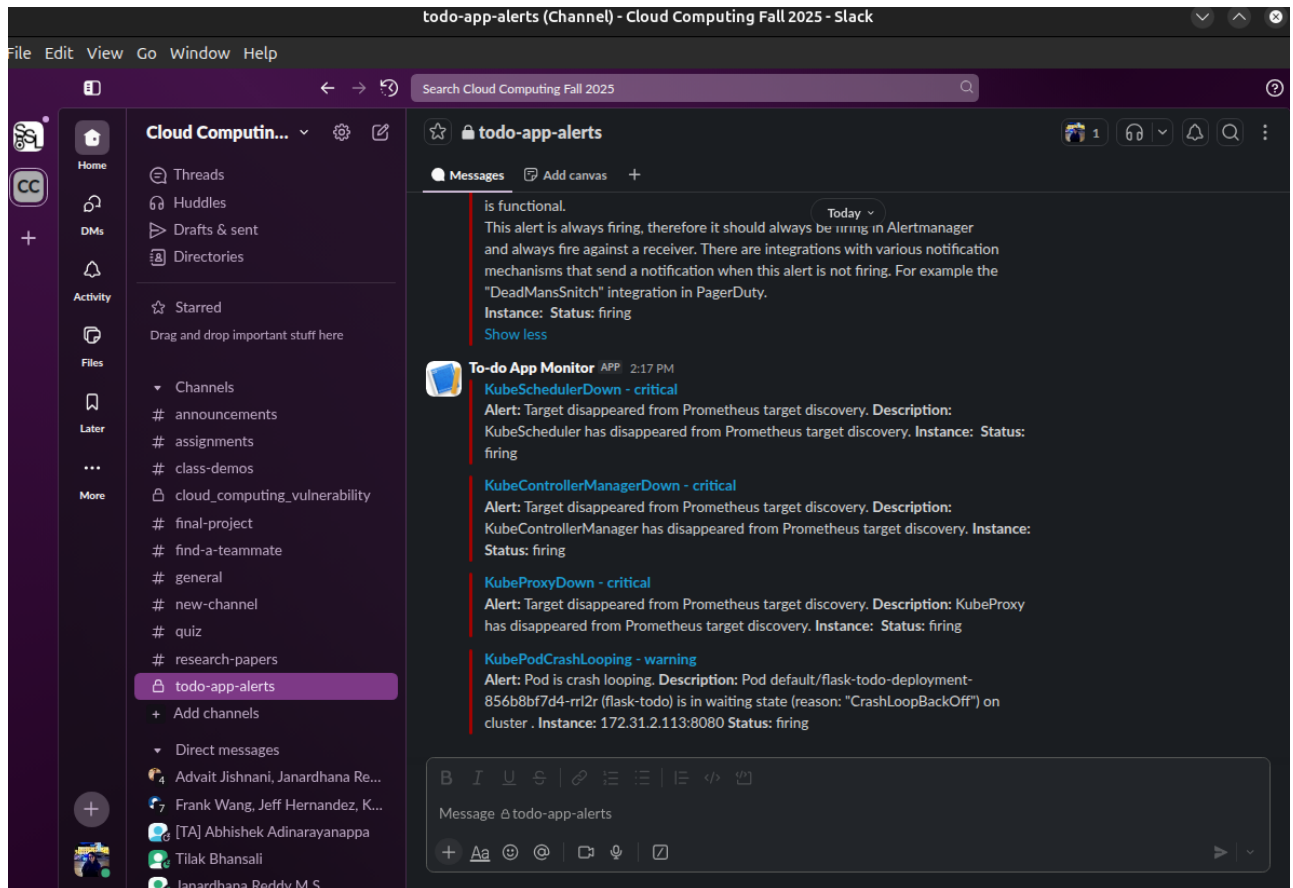
```
kubectl get statefulset -n monitoring
```

NAME	READY	AGE
alertmanager-prometheus-kube-prometheus-alertmanager	1/1	24m
prometheus-prometheus-kube-prometheus-prometheus	1/1	24m

```
kubectl rollout restart statefulset alertmanager-prometheus-kube-prometheus-alertmanager -n monitoring
```

```
kubectl apply -f alert-rule.yaml
prometheusrule.monitoring.coreos.com/flask-health-alert created
```

- Created a new Slack App and added the hook URL in `alertmanager.yaml` which is passed as an environment variable.



- Used the same scenario as of liveness probe to trigger failures. As seen above, a message is sent to alert a pod in crashing loop.
- After re-deploying the flask application, a message is sent to indicate that the event has resolved.

