

Part 1: Creating an Application

1. Download the simple To-Do web application using Flask and MongoDB.
 2. In app.py change host to 0.0.0.0 and add Werkzeug==2.2.2 to the requirements.txt file

Part 2: Containerizing the Application on Docker

1. Install Docker.(brew)
 2. Use two containers: one for Flask, one for MongoDB.
 3. Write a Dockerfile for the Flask application.
 4. Build the Docker image using Docker CLI. **docker build -t todo_app .**
 5. Write a docker-compose file for local testing.
 6. Push the Docker image to Docker Hub. **docker-compose up**
 7. **Push docker image to docker hub docker login**
 8. **docker tag todo_app sashankrm11/todo_app:latest**
 9. **docker push sashankrm11/todo_app**

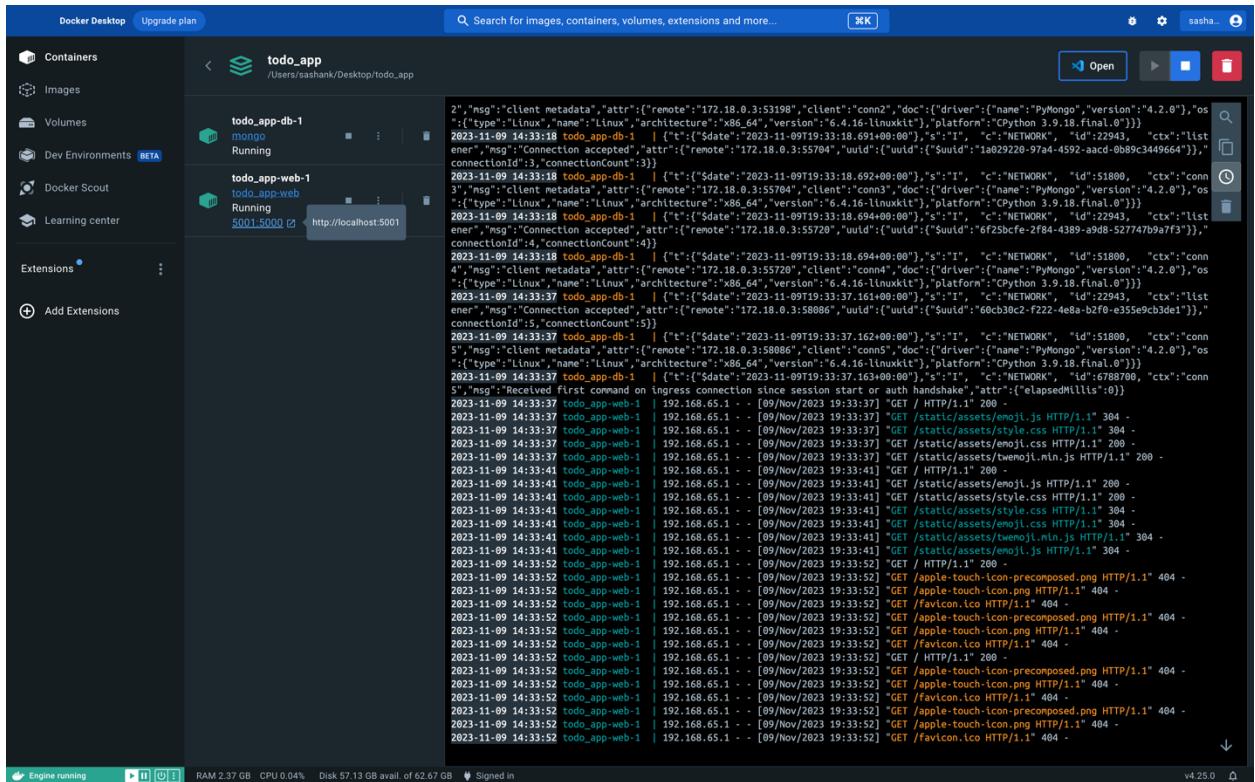
10.

11.

```
*Gracefully stopping... (press Ctrl+C again to force)
Aborting on container exit...
[+/-] Stopping 2/2
✓ Container todo_app-web-1 Stopped
✓ Container todo_app-db-1 Stopped
canceled
[sashank@sashanks-MBP todo_app % docker login
Authenticating with existing credentials...
Login Succeeded
[sashank@sashanks-MBP todo_app % docker tag todo_app sashankrm11/todo_app:latest

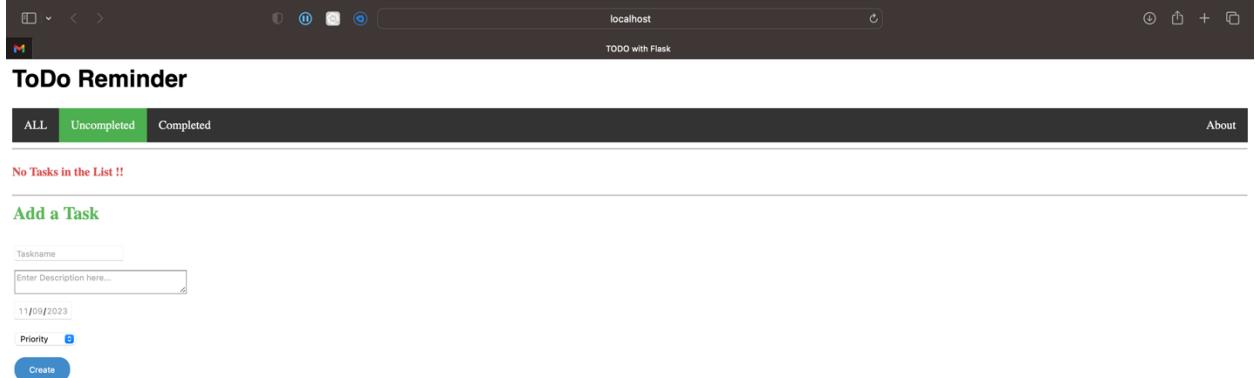
[sashank@sashanks-MBP todo_app % docker push sashankrm11/todo_app

Using default tag: latest
The push refers to repository [docke.io/sashankrm11/todo_app]
b05531b40ef3: Pushed
6bf46d66f708: Pushed
79d209164f32: Pushed
d2ae857413fc: Mounted from library/python
3e71elbda445: Pushed
abc4ad0d31b0: Mounted from library/python
87579ac672ec: Mounted from library/python
ec983b166360: Mounted from library/python
latest: digest: sha256:b974d52ba989badc7f8c5f2d4105cd34e2d749dd50f0ff300debe2c1faed951 size: 199
sashank@sashanks-MBP todo_app %
```

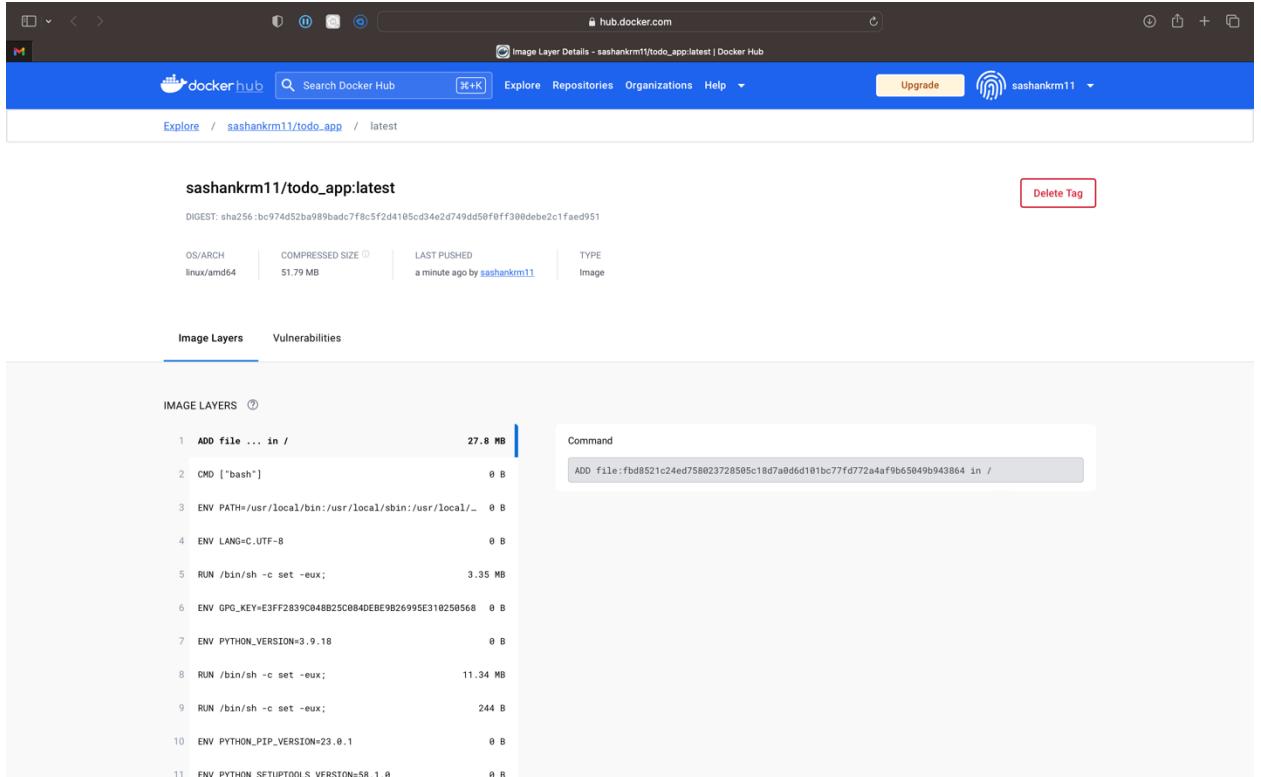


12. Engine running RAM 2.37 GB CPU 0.04% Disk 57.13 GB avail. of 62.67 GB Signed in

13. Check the link provided on docker app or directly go to the localhost with port number.



14.



15.

Part 3: Deploying the Application on Minikube

1. Install Minikube.
2. Create yml file
3. Start Minikube.- **minikube start**
4. Log into docker- **docker login**
5. Create pods for Flask app and MongoDB.
6. Create a Kubernetes deployment for the app.
7. Create and Expose Kubernetes deployment- **kubectl apply -f minikube.yml**
8. Check the pods **kubectl get pods**
9. Test the application on Minikube. **minikube service my-flask-service**

```
sashank@010-18-244-156 todo_app % minikube start
😄  minikube v1.32.0 on Darwin 13.0.1
⚡  Automatically selected the docker driver
📌  Using Docker Desktop driver with root privileges
💡  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
🔥  Creating docker container (CPUs=2, Memory=7797MB) ...
🌐  Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
    └─ Generating certificates and keys ...
    └─ Booting up control plane ...
    └─ Configuring RBAC rules ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
🌐  Verifying Kubernetes components...
    └─ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🚀  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
sashank@010-18-244-156 todo_app % docker login
Authenticating with existing credentials...
Login Succeeded
```

10.

```

minikube
sashank@sashanks-MBP todo_app % kubectl apply -f minikube.yml

persistentvolumeclaim/mongo-pvc created
service/pod1-service unchanged
pod/my-flask-service unchanged
service/my-flask-service configured

11.
sashank@10-18-244-156 todo_app % kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
my-flask-service  0/1   ContainerCreating   0   42s
pod1      0/1   ContainerCreating   0   42s
sashank@10-18-244-156 todo_app % kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
my-flask-service  1/1   Running   0   2m2s
pod1      1/1   Running   0   2m2s

12.
sashank@sashanks-MBP todo_app % minikube service my-flask-service

|-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|-----|
| default | my-flask-service | 56012 | http://192.168.49.2:30427 |
|-----|-----|-----|-----|
💡 Starting tunnel for service my-flask-service.

|-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|-----|
| default | my-flask-service | | http://127.0.0.1:60751 |
|-----|-----|-----|-----|
🎉 Opening service default/my-flask-service in default browser...
❗ Because you are using a Docker driver on darwin, the terminal needs to be open to run it.

13.
^CStopping tunnel for service my-flask-service.



```

ToDo Reminder

To-Do LIST :		Search Reference:	
Status	Task Name	Description Name	Date
<input checked="" type="checkbox"/> apples	get apples		2023-11-12
Add a Task		<input type="text" value="Unique ID"/> <input type="button" value="Search"/>	
Taskname: <input type="text"/> Enter Description here... <input type="text"/> 11/12/2023 <input type="text"/> Priority: <input type="radio"/> <input type="radio"/>		Create 	

14.

15. Now delete the pod and apply again to check if data still persists

- kubectl delete pod pod1
- kubectl apply -f minikube.yml

- minikube service my-flask-service

```
sashank@sashanks-MBP todo_app % kubectl delete pod pod1
pod "pod1" deleted
sashank@sashanks-MBP todo_app % kubectl apply -f minikube.yml
persistentvolumeclaim/mongo-pvc unchanged
pod/pod1 created
service/pod1-service unchanged
pod/my-flask-service unchanged
service/my-flask-service unchanged
sashank@sashanks-MBP todo_app % minikube service my-flask-service

|-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|-----|
| default | my-flask-service | 56012 | http://192.168.49.2:30427 |
|-----|-----|-----|-----|
🏃 Starting tunnel for service my-flask-service.

|-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|-----|
| default | my-flask-service | | http://127.0.0.1:60700 |
|-----|-----|-----|-----|
🎉 Opening service default/my-flask-service in default browser...
❗ Because you are using a Docker driver on darwin, the terminal needs to be open to run it.
^C✋ Stopping tunnel for service my-flask-service.
```

To Do Reminder

ALL Uncompleted Completed

Search Reference: Unique ID Search

Status	Task Name	Description Name	Date	Priority	Remove	Modify
X	apples	get apples	2023-11-12	High !!!		

Add a Task

Taskname:
Enter Description here...
11/12/2023
Priority:
Create

16. Also check the minikube dashboard- **minikube dashboard**

```
sashank@10-18-244-156 todo_app % minikube dashboard
└─ Enabling dashboard ...
  └─ Using image docker.io/kubernetesui/dashboard:v2.7.0
  └─ Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
  └─ Some dashboard features require the metrics-server addon. To enable all features please run:
      minikube addons enable metrics-server

  └─ Verifying dashboard health ...
  └─ Launching proxy ...
  └─ Opening http://127.0.0.1:65236/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
/var/folders/7d/2v6xh03n0tv3rcj_z2tj89p0000gn/T/TemporaryItems/NSIRD_screencaptureui_fUIJ0z/Screenshot\ 2023-11-09\ at\ 4.19.48\ PM.png
```

17.

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
my-flask-service	sashankrm11/todo_app:latest	app: my-flask-service	minikube	Running	0	-	-	4 minutes ago
pod1	mongo	name: pod1	minikube	Running	0	-	-	4 minutes ago

18.

The screenshot shows the Kubernetes Dashboard interface. The left sidebar has a 'Services' section selected. The main content area displays a table of services:

Name	Labels	Type	Cluster IP	Internal Endpoints	External Endpoints	Created
my-flask-service	-	LoadBalancer	10.101.253.200	my-flask-service:56012 TCP my-flask-service:30427 TCP	127.0.0.1:56012	4 minutes ago
pod1-service	-	ClusterIP	10.103.200.183	pod1-service:27017 TCP pod1-service:0 TCP	-	4 minutes ago
kubernetes	component: apiserver provider: kubernetes	ClusterIP	10.96.0.1	kubernetes:443 TCP kubernetes:0 TCP	-	4 minutes ago

19.

20. Persistent volume-

The screenshot shows the Kubernetes Dashboard interface. The left sidebar has a 'Persistent Volumes' section selected. The main content area displays a table of persistent volumes:

Name	Capacity	Access Modes	Reclaim Policy	Status	Claim	Storage Class	Reason	Created
pvc-176135cb-f700-4764-b920-e52e9bb313fc	storage: 10Gi	ReadWriteOnce	Delete	Bound	default/mongo-pvc	standard	-	11 hours ago

Part 4: Deploying the Application on AWS EKS

1. Create an AWS EKS cluster. - **eksctl create cluster --name todo-cluster --version 1.23 --region us-west-2 --nodegroup-name amazon-nodes --node-type t3.medium --nodes 3 --nodes-min 1 --nodes-max 4 --managed**

```

~/Users/sashank/Desktop/todo_app -- zsh          ~| ~/Desktop/todo_app -- ssh - minikube tunnel ... | ~/Desktop/todo_app -- kubectl + minikube dashboard + Done
Q: cloud-assignment3files-db-1

sashank@sashanks-MBP:~/Desktop/todo_app% eksctl create cluster --name todo-cluster --version 1.23 --region us-west-2
--nodegroup-name amazon-nodes --node-type t3.medium --nodes 3 --nodes-min 1
--nodes-max 4 --managed

2023-11-18 13:45:03 [!] eksctl version 0.164.0-dev-3cda1af9e.2023-10-27T12:27:01Z
2023-11-18 13:45:03 [!] using region us-west-2
2023-11-18 13:45:04 [!] setting availability zones to [us-west-2a us-west-2d us-west-2c]
2023-11-18 13:45:04 [!] subnets for us-west-2a - public:192.168.0.0/19 private:192.168.96.0/19
2023-11-18 13:45:04 [!] subnets for us-west-2d - public:192.168.32.0/19 private:192.168.128.0/19
2023-11-18 13:45:04 [!] subnets for us-west-2c - public:192.168.64.0/19 private:192.168.160.0/19
2023-11-18 13:45:04 [!] nodegroup "ng-20fbeafc" will use "" [AmazonLinux2/1.23]
2023-11-18 13:45:04 [!] using Kubernetes version 1.23
2023-11-18 13:45:04 [!] creating EKS cluster "todo-cluster" in "us-west-2" region with managed nodes
2023-11-18 13:45:04 [!] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2023-11-18 13:45:04 [!] if you encounter any issues, check CloudFormation console or try `eksctl utils describe-stacks --region=us-west-2 --cluster=todo-cluster`
2023-11-18 13:45:04 [!] Kubernetes API endpoint access will use default or {publicAccess=true, privateAccess=false} for cluster "todo-cluster" in "us-west-2"
2023-11-18 13:45:04 [!] CloudWatch logging will not be enabled for Cluster "todo-cluster" in "us-west-2"
2023-11-18 13:45:04 [!] you can enable it with `eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-west-2 --cluster=todo-cluster`
2023-11-18 13:45:04 [!]
2 sequential tasks: { create cluster control plane "todo-cluster",
  2 sequential sub-tasks: {
    wait for control plane to become ready,
    create managed nodegroup "ng-20fbeafc",
  }
}

2023-11-18 13:45:04 [!] building cluster stack "eksctl-todo-cluster-cluster"
2023-11-18 13:45:06 [!] deploying stack "eksctl-todo-cluster-cluster"
2023-11-18 13:45:36 [!] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
2023-11-18 13:46:08 [!] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
2023-11-18 13:46:10 [!] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
2023-11-18 13:46:10 [!] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
2023-11-18 13:49:10 [!] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
2023-11-18 13:49:10 [!] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
2023-11-18 13:50:11 [!] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
2023-11-18 13:51:12 [!] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
2023-11-18 13:52:12 [!] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
2023-11-18 13:53:13 [!] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
2023-11-18 13:53:13 [!] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
2023-11-18 13:55:14 [!] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
2023-11-18 13:55:14 [!] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
2023-11-18 14:04:08 [!] deploying stack "eksctl-todo-cluster-nodegroup-ng-20fbeafc"
2023-11-18 14:04:08 [!] waiting for CloudFormation stack "eksctl-todo-cluster-nodegroup-ng-20fbeafc"
2023-11-18 14:04:40 [!] waiting for CloudFormation stack "eksctl-todo-cluster-nodegroup-ng-20fbeafc"
2023-11-18 14:05:14 [!] waiting for CloudFormation stack "eksctl-todo-cluster-nodegroup-ng-20fbeafc"
2023-11-18 14:06:14 [!] waiting for CloudFormation stack "eksctl-todo-cluster-nodegroup-ng-20fbeafc"
2023-11-18 14:06:16 [!] saving kubeconfig to "/Users/sashank/.kube/config"
2023-11-18 14:06:16 [!] no tasks
2023-11-18 14:06:16 [!] all EKS cluster resources for "todo-cluster" have been created
2023-11-18 14:06:17 [!] nodegroup "ng-20fbeafc" has 2 node(s)
2023-11-18 14:06:17 [!] node "ip-192-168-24-204.us-west-2.compute.internal" is ready
2023-11-18 14:06:17 [!] node "ip-192-168-32-170.us-west-2.compute.internal" is ready
2023-11-18 14:06:17 [!] waiting for node(s) to become ready in "ng-20fbeafc"
2023-11-18 14:06:17 [!] nodegroup "ng-20fbeafc" has 2 node(s)
2023-11-18 14:06:17 [!] node "ip-192-168-24-204.us-west-2.compute.internal" is ready
2023-11-18 14:06:17 [!] node "ip-192-168-32-170.us-west-2.compute.internal" is ready
2023-11-18 14:06:19 [!] kubelet command should work with "/Users/sashank/.kube/config", try `kubectl get nodes`
2023-11-18 14:06:19 [!] EKS cluster "todo-cluster" in "us-west-2" region is ready
zsh: command not found: --nodegroup-name
zsh: command not found: --nodes-max

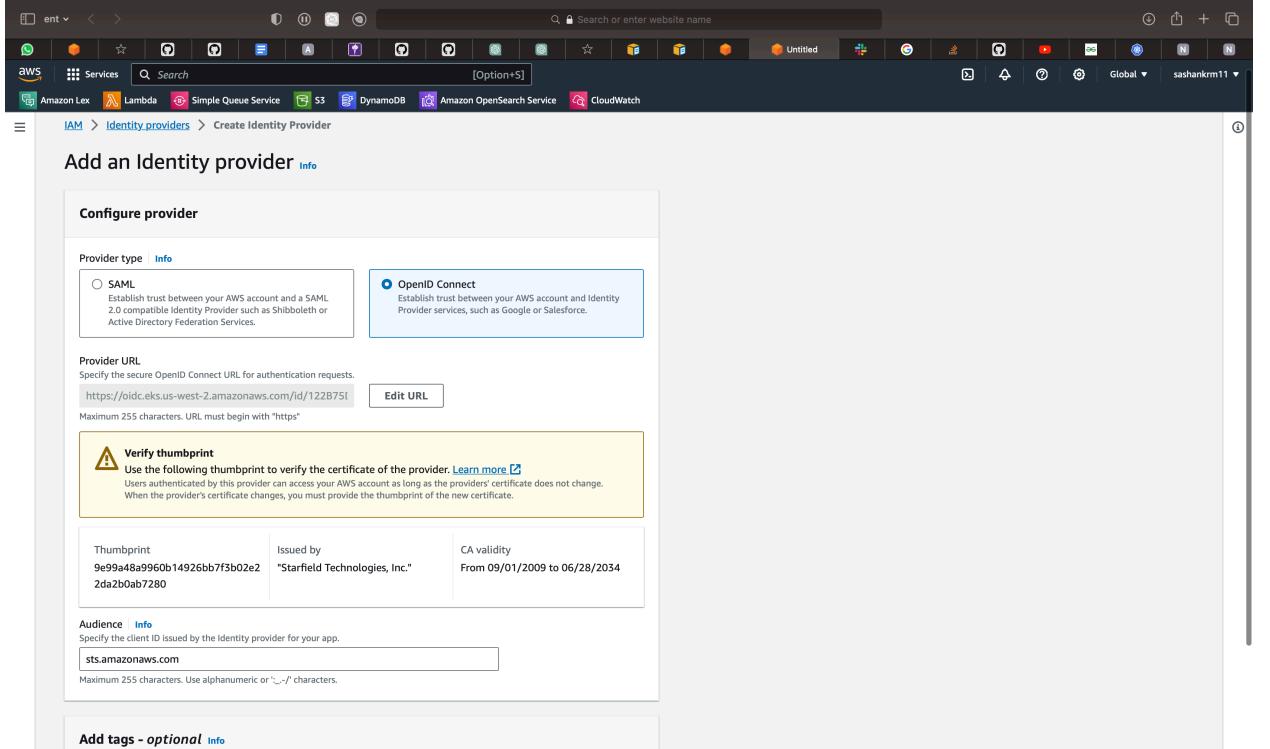
```

2. Configure kubectl for EKS. **aws eks --region us-west-2 update-kubeconfig --name todo-cluster**

3. Create an ecr image (follow steps given in aws console)-

- **aws ecr get-login-password --region us-west-2 | docker login --username AWS - -password-stdin 475338120318.dkr.ecr.us-west-2.amazonaws.com**
- **docker build -t todo-flask-app .**
- **docker tag todo-flask-app:latest 475338120318.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:latest**
- **docker push 475338120318.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app**

4. sashank@sashanks-MBP todo_app %
 5. Set up iam roles for EKS to interact with Elastic Block Store (EBS) and make sure you finally get the **AmazonEKS_EBS_CSI_Driver**
 - 6.



The screenshot shows the AWS EKS Cluster Details page for a cluster named 'todo-cluster'. The 'Networking' tab is selected. A prominent message at the top states: 'Your cluster's Kubernetes version (1.23) is on extended support until October 11, 2024. To opt out of extended support, upgrade your cluster to the next supported Kubernetes version.' Below this, the 'Cluster info' section shows the status as 'Active', Kubernetes version as '1.23', support type as 'Extended support until October 11, 2024', and provider as 'EKS'. A 'Feedback Survey' overlay is present. The 'Add-ons' section lists the 'Amazon EBS CSI Driver' add-on, which is enabled for storage and has a version of v1.24.1-eksbuild.1. The page also includes tabs for Overview, Resources, Compute, Networking, Add-ons, Authentication, Observability, Update history, and Tags.

7. Create a Kubernetes deployment in EKS.
 - **kubectl apply -f mongodb-pvc.yaml**
 - **kubectl apply -f mongodb-deployment.yaml**
 - **kubectl apply -f mongodb-service.yaml**
 - **kubectl apply -f flask-deployment.yaml**
 - **kubectl apply -f flask-service.yaml**

8. check the pods and service **kubectl get pods , kubectl get services, kubectl get pvc**

```
release todo-dev uninstalled
[sashank@sashanks-MBP todo_app % kubectl apply -f mongodb-pvc.yaml
persistentvolumeclaim/mongo-pvc created
[sashank@sashanks-MBP todo_app % kubectl apply -f mongodb-deployment.yaml
deployment.apps/pod1-deployment created
[sashank@sashanks-MBP todo_app % kubectl apply -f mongodb-service.yaml
service/pod1-service created
[sashank@sashanks-MBP todo_app % kubectl apply -f flask-deployment.yaml
deployment.apps/my-flask-service-deployment created
[sashank@sashanks-MBP todo_app % kubectl apply -f flask-rc.yaml
replicationcontroller/my-flask-app-rc created
[sashank@sashanks-MBP todo_app % kubectl apply -f flask-service.yaml
service/my-flask-service created
```

- 9.

```
sashank@sashanks-MBP todo_app % kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
my-flask-service-deployment-556d9755d4-4smf7  1/1     Running   0          58m
my-flask-service-deployment-556d9755d4-5zvkw  1/1     Running   0          58m
my-flask-service-deployment-556d9755d4-67ig2  1/1     Running   0          58m
my-flask-service-deployment-556d9755d4-7vgrd  1/1     Running   0          58m
my-flask-service-deployment-556d9755d4-dfnff  1/1     Running   0          58m
my-flask-service-deployment-556d9755d4-zzdh6  1/1     Running   0          58m
pod1-deployment-7cc4694ddd-9zcrt  1/1     Running   0          58m
sashank@sashanks-MBP todo_app % kubectl get services
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP
kubernetes   ClusterIP   10.100.0.1   <none>
my-flask-service   LoadBalancer  10.100.224.232  aa4dba753295c48fea738acc5cc7b575-293773832.us-west-2.elb.amazonaws.com
pod1-service   ClusterIP   10.100.135.68  <none>
sashank@sashanks-MBP todo_app %
```

- 10.

```
[sashank@sashanks-MBP todo_app % kubectl get pvc
NAME      STATUS    VOLUME          CAPACITY  ACCESS MODES  STORAGECLASS  AGE
mongo-pvc  Bound     pvc-14315eca-ef2e-4e63-85fc-4e73092bb013  10Gi       RWO          gp2           21m
```

11. Test the application on AWS EKS. <http://af3b62136dba94e1fdb8b67e078c6d46-1217243179.us-west-2.elb.amazonaws.com:56011/list>

12.

13. Now delete the pod and check if the data still persists

```
my-flask-service-deployment-79b5669ff4-66ck9  1/1   Running  0   22m
my-flask-service-deployment-79b5669ff4-7266k  1/1   Running  0   22m
my-flask-service-deployment-79b5669ff4-c5617  1/1   Running  0   22m
my-flask-service-deployment-79b5669ff4-mdsz5  1/1   Running  0   22m
my-flask-service-deployment-79b5669ff4-ptfzc  1/1   Running  0   22m
my-flask-service-deployment-79b5669ff4-zkg8v  1/1   Running  0   22m
pod1-deployment-84ddf654bf-46mxz            1/1   Running  0   22m
[sashank@sashanks-MBP todo_app % kubectl delete pod pod1-deployment-84ddf654bf-46mxz
pod "pod1-deployment-84ddf654bf-46mxz" deleted
[sashank@sashanks-MBP todo_app % kubectl get pods
  NAME                READY  STATUS    RESTARTS  AGE
  my-flask-app-rc-9jjhw  1/1   Running  0   26m
  my-flask-app-rc-bwbqj  1/1   Running  0   26m
  my-flask-app-rc-fzdz7  1/1   Running  0   26m
  my-flask-app-rc-p7716  1/1   Running  0   26m
  my-flask-app-rc-v455n  1/1   Running  0   26m
  my-flask-app-rc-xj8tr  1/1   Running  0   26m
  my-flask-service-deployment-79b5669ff4-66ck9  1/1   Running  0   26m
  my-flask-service-deployment-79b5669ff4-7266k  1/1   Running  0   26m
  my-flask-service-deployment-79b5669ff4-c5617  1/1   Running  0   26m
  my-flask-service-deployment-79b5669ff4-mdsz5  1/1   Running  0   26m
  my-flask-service-deployment-79b5669ff4-ptfzc  1/1   Running  0   26m
  my-flask-service-deployment-79b5669ff4-zkg8v  1/1   Running  0   26m
  pod1-deployment-84ddf654bf-9k65b            1/1   Running  0   5s
```

14.

To-Do Reminder

ALL Uncompleted Completed About

Search Reference: Unique ID Search

To-Do LIST :

Status	Task Name	Description Name	Date	Priority	Remove	Modify
X oranges	get orange		11/12/2023	High !!!		

Add a Task v2

Taskname:
Enter Description here...

11/12/2023

Priority:

15.

Part 5: Replication Controller Feature

1. Create a replication controller yml file.
2. Set and modify the number of replicas.
3. Use kubectl to manage the replication controller. **kubectl apply -f flask-rc.yml**
4. Test the replication controller's functionality.
 - **kubectl get pods -l app=my-flask-rc-app**
 - **kubectl delete pod -l app=my-flask-rc-app**
 - **kubectl get pods -l app=my-flask-rc-app (see that pods restarts)**
 - **kubectl get rc my-flask-app-rc**

```

~/Users/sashank/Desktop/todo_app -- zsh
~/Desktop/todo_app -- ssh - minikube tunnel ...
~/Desktop/todo_app -- kubectl - minikube dashboard + Done
Q: cloud-assignment/Deployment-1
NAME          READY   STATUS    RESTARTS   AGE
my-flask-service-deployment-55dd9755d4-4smf7  1/1    Running   0          58m
my-flask-service-deployment-55dd9755d4-5zvkw  1/1    Running   0          58m
my-flask-service-deployment-55dd9755d4-67lg2  1/1    Running   0          58m
my-flask-service-deployment-55dd9755d4-7vprd  1/1    Running   0          58m
my-flask-service-deployment-55dd9755d4-dfrff  1/1    Running   0          58m
my-flask-service-deployment-55dd9755d4-zzh6h  1/1    Running   0          58m
pod1-deployment-7c7494ddd-pxzrt   1/1    Running   0          58m
sashank@sashanks-MBP todo_app % kubectl get services
NAME        TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes   ClusterIP   10.100.0.1      <none>           443/TCP      79m
my-flask-service   LoadBalancer   10.100.224.232  aa4d0ba753295c48fea738acc5cc7b575-293773832.us-west-2.elb.amazonaws.com  56811:32268/TCP  58m
pod1-service   ClusterIP   10.100.138.68  <none>           27017/TCP    58m
sashank@sashanks-MBP todo_app % kubectl apply -f flask-rc.yaml

error: the path "flask-rc.yaml" does not exist
sashank@sashanks-MBP todo_app % kubectl apply -f flask-rc.yaml
replicationcontroller/my-flask-app-rc created
sashank@sashanks-MBP todo_app % kubectl get pods -l app=my-flask-rc-app

NAME          READY   STATUS    RESTARTS   AGE
my-flask-app-rc-29bf6  1/1    Running   0          8s
my-flask-app-rc-8mxjz  1/1    Running   0          8s
my-flask-app-rc-9jnxm  1/1    Running   0          8s
my-flask-app-rc-m2vkh  1/1    Running   0          8s
my-flask-app-rc-n5jrm  1/1    Running   0          8s
my-flask-app-rc-nb5xj  1/1    Running   0          8s
sashank@sashanks-MBP todo_app % kubectl delete pod -l app=my-flask-rc-app

pod "my-flask-app-rc-29bf6" deleted
pod "my-flask-app-rc-8mxjz" deleted
pod "my-flask-app-rc-9jnxm" deleted
pod "my-flask-app-rc-m2vkh" deleted
pod "my-flask-app-rc-n5jrm" deleted
pod "my-flask-app-rc-nb5xj" deleted
[14]
sashank@sashanks-MBP todo_app % kubectl get pods -l app=my-flask-rc-app

NAME          READY   STATUS    RESTARTS   AGE
my-flask-app-rc-c74gc  1/1    Running   0          8s
my-flask-app-rc-j5m7s  1/1    Running   0          7s
my-flask-app-rc-icm2nzs 1/1    Running   0          8s
my-flask-app-rc-1802m  1/1    Running   0          8s
my-flask-app-rc-imphj  1/1    Running   0          16s
my-flask-app-rc-qnvk9  1/1    Running   0          15s
my-flask-app-rc-wjkbj  1/1    Running   0          7s
sashank@sashanks-MBP todo_app % kubectl get rc my-flask-app-rc

NAME      DESIRED   CURRENT   READY   AGE
my-flask-app-rc  6        6        6        62s
sashank@sashanks-MBP todo_app %

```

Part 6: Rolling Update Strategy

1. Configure deployment for rolling updates.

- Update the index.html file to show **Add a Task v2** in the index.html file and Build a new docker image for the updated project and push to ecr and update the deployment.yml file with this image url
- **docker build -t 475338120318.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v7 .**
- **docker push 475338120318.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v7**

2. Set and execute the rolling update strategy.

- **kubectl apply -f deployment.yaml**
- **kubectl rollout status deployment/my-flask-service-deployment**
- **kubectl get services**
- **kubectl get pods -l app=my-flask-service**

```

sashank@sashanks-MBP todo_app % docker build -t 475338120318.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v7 .
[+] Building 9.6s (9/9) FINISHED
--> [internal] load build definition from Dockerfile
--> [internal] transfer context: 2B
--> [internal] load build definition from Dockerfile
--> [internal] transfer dockerfile: 223B
--> [internal] transfer manifest: python:3.9-slim
--> [internal] transfer manifest: python:3.9-slim@sha256:2b767480bb794c2999e8555a14bf4e8a50e0cf74cd7b8c613ed4f4a10fb12bf
--> [internal] load build context: 229.9KB
--> [internal] transfer context: 229.9KB
--> [3/4] COPY . /app
--> [4/4] RUN pip install --trusted-host pypi.python.org -r requirements.txt
--> exporting to image
--> writing manifest to image
--> writing image sha256:1936997cd6e012bhddead54bc158e9a8eeae5f40193502ca9e9523e7a6cc75aa1
--> naming to 475338120318.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v7

What's New?
View a summary of image vulnerabilities and recommendations > docker scan quickview
sashank@sashanks-MBP todo_app % docker push 475338120318.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v7
The push refers to a repository [475338120318.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app]
37ba5cb35031: Pushed
25d8b23d1ef1: Pushed
79d209164f32: Layer already exists
d2a69423d1ef1: Layer already exists
3a7e191464f1: Layer already exists
abc4ad0d31b6: Layer already exists
87579ac672ec: Layer already exists
ec983b166366: Layer already exists
v7: digest: sha256:60ffrr4d8e0892c2b73721254beec1fc5f4b9c1a722eed4a6fac5582488fa8a2 size: 1998

```

```

[sashank@sashanks-MBP todo_app % kubectl apply -f flask-deployment.yaml
deployment.apps/my-flask-service-deployment configured
[sashank@sashanks-MBP todo_app % kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
my-flask-app-rc-9jjhw   1/1    Running   0          12h
my-flask-app-rc-bwbqj   1/1    Running   0          12h
my-flask-app-rc-fzd7   1/1    Running   0          12h
my-flask-app-rc-p7716   1/1    Running   0          12h
my-flask-app-rc-v455n   1/1    Running   0          12h
my-flask-app-rc-xj8tr   1/1    Running   0          12h
my-flask-service-deployment-79b5669ff4-c5617 1/1    Running   0          12h
my-flask-service-deployment-964b4cb49-99pwf 1/1    Running   0          21s
my-flask-service-deployment-964b4cb49-fkgbf  0/1   ContainerCreating   0          1s
my-flask-service-deployment-964b4cb49-ftln4  1/1    Running   0          42s
my-flask-service-deployment-964b4cb49-h15fg  0/1   ContainerCreating   0          1s
my-flask-service-deployment-964b4cb49-k9x5q  1/1    Running   0          21s
my-flask-service-deployment-964b4cb49-pmqnc  1/1    Running   0          42s
pod1-deployment-84dd654bf-9k65b   1/1    Running   0          11h
[sashank@sashanks-MBP todo_app % kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
my-flask-app-rc-9jjhw   1/1    Running   0          12h
my-flask-app-rc-bwbqj   1/1    Running   0          12h
my-flask-app-rc-fzd7   1/1    Running   0          12h
my-flask-app-rc-p7716   1/1    Running   0          12h
my-flask-app-rc-v455n   1/1    Running   0          12h
my-flask-app-rc-xj8tr   1/1    Running   0          12h
my-flask-service-deployment-964b4cb49-99pwf 1/1    Running   0          51s
my-flask-service-deployment-964b4cb49-fkgbf  1/1    Running   0          31s
my-flask-service-deployment-964b4cb49-ftln4  1/1    Running   0          72s
my-flask-service-deployment-964b4cb49-h15fg  1/1    Running   0          31s
my-flask-service-deployment-964b4cb49-k9x5q  1/1    Running   0          51s
my-flask-service-deployment-964b4cb49-pmqnc  1/1    Running   0          72s
pod1-deployment-84dd654bf-9k65b   1/1    Running   0          11h
[sashank@sashanks-MBP todo_app % kubectl rollout status deployment/my-flask-service-deployment
deployment "my-flask-service-deployment" successfully rolled out
[sashank@sashanks-MBP todo_app % kubectl get services
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes     ClusterIP   10.100.0.1    <none>           443/TCP         2d1h
my-flask-service LoadBalancer 10.100.82.228  af3b62136dba94e1fdb8b67e078c6d46-1217243179.us-west-2.elb.amazonaws.com 56811:38954/TCP 12h
pod1-service   ClusterIP   10.100.86.219  <none>           27017/TCP       12h
[sashank@sashanks-MBP todo_app %

```

3. Test the updated application.(check if we can see Add a Task v3 in index.html file)

To-Do LIST :		Description Name		Date	Priority	Remove	Modify
Status	Task Name						
<input checked="" type="checkbox"/>	oranges	get orange			High !!!	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>

Add a Task v3

Taskname:

Enter Description here...

11/12/2023

Priority:

Part 7: Health Monitoring

1. Set up liveness and readiness probes.
2. Configure Kubernetes to respond to probe failures.
3. Monitor pod health.
4. Test health monitoring by inducing failures.
5. Health Failure- add this to deployment file

- env:
 - name: LIVENESS_PROBE_FAIL

value: "true"
- kubectl apply -f deployment.yaml
- kubectl get pods
- kubectl describe pod my-flask-service-deployment-6cc59cdcb7-7k59t


```
[sashank@sashanks-MBP todo_app % kubectl apply -f flask-deployment.yaml
deployment.apps/my-flask-service-deployment configured
[sashank@sashanks-MBP todo_app % kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
my-flask-app-rc-c74gc              1/1    Running   0          24h
my-flask-app-rc-j5m7s              1/1    Running   0          24h
my-flask-app-rc-l8n2m              1/1    Running   0          24h
my-flask-app-rc-lmpjh              1/1    Running   0          24h
my-flask-app-rc-qnvk9              1/1    Running   0          24h
my-flask-app-rc-wjkjbj             1/1    Running   0          24h
my-flask-service-deployment-6cc59cdcb7-6xns4  1/1    Running   1 (32s ago) 93s
my-flask-service-deployment-6cc59cdcb7-7k59t  0/1    Running   1 (12s ago) 73s
my-flask-service-deployment-6cc59cdcb7-1frl6  1/1    Running   1 (32s ago) 93s
my-flask-service-deployment-6cc59cdcb7-rxr75  1/1    Running   1 (53s ago) 114s
my-flask-service-deployment-6cc59cdcb7-tpq5s   0/1    Running   1 (12s ago) 72s
my-flask-service-deployment-6cc59cdcb7-wh6lr   1/1    Running   1 (53s ago) 114s
pod1-deployment-7cc4694ddd-9zcr7  1/1    Running   0          25h
[sashank@sashanks-MBP todo_app % kubectl logs my-flask-service-deployment-6cc59cdcb7-7k59t
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://192.168.17.92:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 670-025-399
192.168.24.204 - - [11/Nov/2023 20:55:48] "GET /ready HTTP/1.1" 200 -
192.168.24.204 - - [11/Nov/2023 20:55:48] "GET /health HTTP/1.1" 500 -
Events:
  Type  Reason  Age            From           Message
  Normal Scheduled  2ms           default-scheduler  Successfully assigned default/my-flask-service-deployment-6cc59cdcb7-7k59t to ip-192-168-24-204.us-west-2.compute.internal
  Normal Pulled   2s            kubelet        Successfully pulled image "47538128938.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v6" in 136.82765ms (136.840545ms including waiting)
  Normal Pulling  69s (x2 over 2m) kubelet        Pulling image "47538128938.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v6"
  Normal Created  69s (x2 over 2m) kubelet        Created container todo
  Normal Started  69s (x2 over 2m) kubelet        Started container todo
  Normal Killing  69s            kubelet        Container todo failed liveness probe, will be restarted
  Normal Pulled   69s            kubelet        Successfully pulled image "47538128938.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v6" in 134.481447ms (134.494308ms including waiting)
  Warning Unhealthy 29s (x5 over 100s) kubelet        Liveness probe failed: HTTP probe failed with statuscode: 500
```

6. Readiness Failure- add this to deployment file

- env:
 - name: READINESS_PROBE_FAIL

value: "true"
- kubectl apply -f deployment.yaml
- kubectl get pods
- kubectl describe pod my-flask-service-deployment-645fd877c8-xxtkp

```

[sashank@sashanks-MBP todo_app % kubectl apply -f flask-deployment.yaml
deployment.apps/my-flask-service-deployment configured
[sashank@sashanks-MBP todo_app % kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
my-flask-app-rc-c74gc      1/1    Running   0          24h
my-flask-app-rc-j5m7s      1/1    Running   0          24h
my-flask-app-rc-l8n2m      1/1    Running   0          24h
my-flask-app-rc-lmpjh      1/1    Running   0          24h
my-flask-app-rc-qnvk9      1/1    Running   0          24h
my-flask-app-rc-wjkjbj     1/1    Running   0          9s
my-flask-service-deployment-645fd877c8-8fjfvy  0/1    Running   0          9s
my-flask-service-deployment-645fd877c8-xxtkp     0/1    Running   0          9s
my-flask-service-deployment-6cc59cdcb7-6xns4      0/1    CrashLoopBackOff  5 (33s ago)  6m34s
my-flask-service-deployment-6cc59cdcb7-lfrl6      0/1    CrashLoopBackOff  5 (34s ago)  6m34s
my-flask-service-deployment-6cc59cdcb7-rxr75      0/1    CrashLoopBackOff  5 (54s ago)  6m55s
my-flask-service-deployment-6cc59cdcb7-tpq5s       0/1    CrashLoopBackOff  5 (13s ago)  6m13s
my-flask-service-deployment-6cc59cdcb7-wh6lr       0/1    CrashLoopBackOff  5 (54s ago)  6m55s
pod1-deployment-7cc4694ddd-9z crt                 1/1    Running   0          25h
[sashank@sashanks-MBP todo_app % kubectl logs my-flask-service-deployment-645fd877c8-xxtkp
  * Serving Flask app 'app' (lazy loading)
  * Environment: production
    WARNING: This is a development server. Do not use it in a production deployment.
    Use a production WSGI server instead.
  * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
  * Running on all addresses (0.0.0.0)
  * Running on http://127.0.0.1:5000
  * Running on http://192.168.6.239:5000
Press CTRL+C to quit
  * Restarting with stat
  * Debugger is active!
  * Debugger PIN: 100-269-977
192.168.24.204 - - [11/Nov/2023 21:00:52] "GET /ready HTTP/1.1" 500 -
192.168.24.204 - - [11/Nov/2023 21:00:52] "GET /health HTTP/1.1" 200 -
[sashank@sashanks-MBP todo_app % kubectl describe pod my-flask-service-deployment-645fd877c8-xxtkp
Name:           my-flask-service-deployment-645fd877c8-xxtkp
Namespace:      default
Priority:       0
Service Account: default
Node:          ip-192-168-24-204.us-west-2.compute.internal/192.168.24.204
Start Time:     Sat, 11 Nov 2023 16:00:32 -0500
Labels:         app=my-flask-service
                pod-template-hash=645fd877c8
Annotations:   kubernetes.io/psp: eks.privileged
Status:        Running
IP:            192.168.6.239
Events:
  Type  Reason  Age From      Message
  ----  -----  --  --        --
  Normal Scheduled  49s default-scheduler  Successfully assigned default/my-flask-service-deployment-645fd877c8-xxtkp to ip-192-168-24-204.us-west-2.compute.internal
  Normal Pulling   39s kubelet      Pulling image "475338120318.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v6"
  Normal Pulled   39s kubelet      Successfully pulled image "475338120318.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v6" in 129.617043ms (129.622332ms including waiting)
  Normal Created   39s kubelet      Created container todo
  Normal Started   39s kubelet      Started container todo
  Warning Unhealthy  28s kubelet      Readiness probe failed: HTTP probe failed with statuscode: 500

```

Part 8: Alerting (Extra Credit)

1. Set up Prometheus
 - brew install helm
 - helm repo add prometheus-community <https://prometheus-community.github.io/helm-charts>
 - helm repo update
 - helm install todo-dev prometheus-community/kube-prometheus-stack

```

[sashank@sashanks-MBP todo_app % brew install helm
Running `brew update --auto-update`...
==> Auto-updated Homebrew!
Updated 1 tap (homebrew/core).
==> New Formulae
chainsaw

cliam

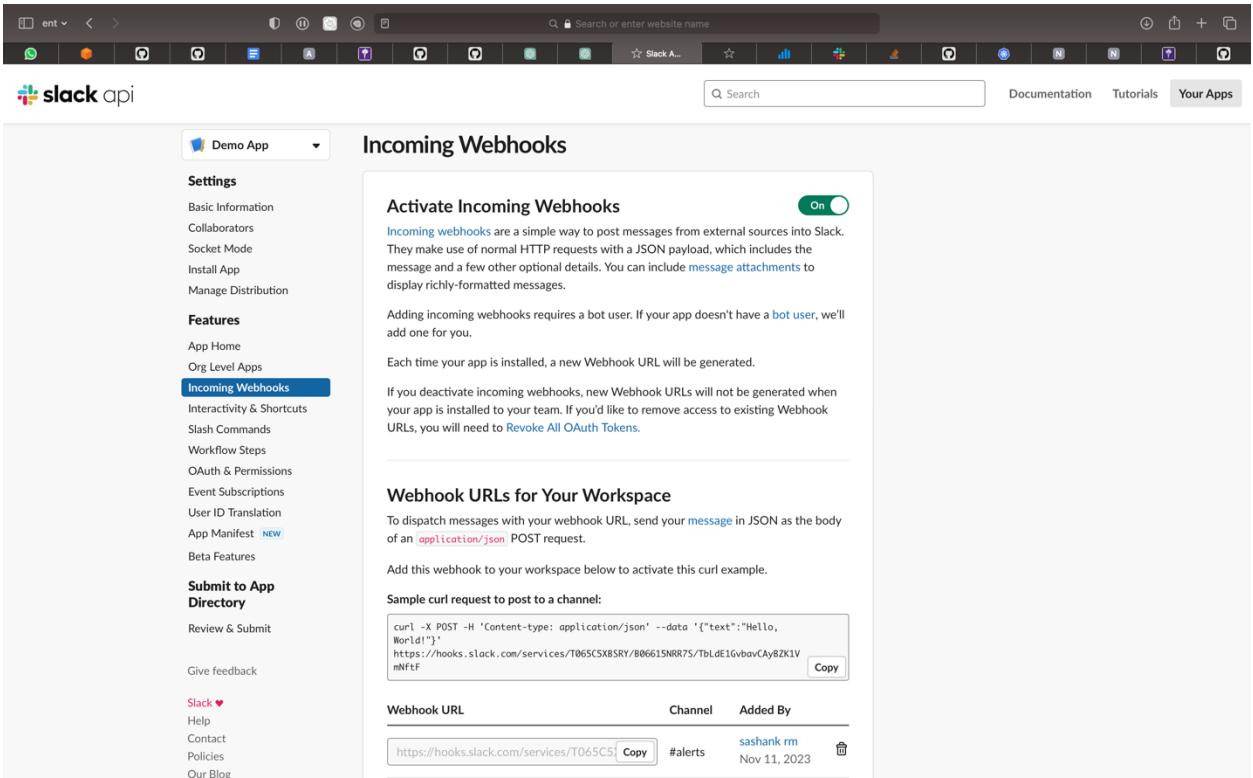
You have 12 outdated formulae installed.

==> Downloading https://ghcr.io/v2/homebrew/core/helm/manifests/3.13.2
#####
==> Fetching helm
==> Downloading https://ghcr.io/v2/homebrew/core/helm/blobs/sha256:c6d27896985706bb3b3f6ffec77ed0ca57552a02d98211c6
#####
==> Pouring helm--3.13.2.ventura.bottle.tar.gz
==> Caveats
zsh completions have been installed to:
  /usr/local/share/zsh/site-functions
==> Summary
  /usr/local/Cellar/helm/3.13.2: 65 files, 53.8MB
==> Running `brew cleanup helm`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
sashank@sashanks-MBP todo_app % helm repo add prometheus-community https://prometheus-community.github.io/helm-char

"prometheus-community" has been added to your repositories
[sashank@sashanks-MBP todo_app % 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 Helming!*
[sashank@sashanks-MBP todo_app % helm install todo-dev prometheus-community/kube-prometheus-stack
NAME: todo-dev
LAST DEPLOYED: Sun Nov 12 01:09:00 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
kube-prometheus-stack has been installed. Check its status by running:
  kubectl --namespace default get pods -l "release=todo-dev"
- Visit https://github.com/prometheus-operator/kube-prometheus for instructions on how to create & configure Alertmanager and Prometheus instances using the Operator.

```

2. Create a Slack App and generate a webhook URL,
<https://hooks.slack.com/services/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>



The screenshot shows the Slack App configuration interface. The left sidebar includes sections for Settings, Features, Incoming Webhooks (which is currently selected), Submit to App Directory, and Slack ♥. The main content area is titled 'Incoming Webhooks' and contains the following information:

- Activate Incoming Webhooks:** A toggle switch labeled 'On' is shown. Below it, text explains that incoming webhooks allow posting messages from external sources into Slack via normal HTTP requests with a JSON payload.
- Webhook URLs for Your Workspace:** Instructions for sending messages with a curl example and a table of existing webhook URLs.

Webhook URL	Channel	Added By
https://hooks.slack.com/services/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	#alerts	sashank rm Nov 11, 2023

3. Modify the alertmanager config to add the webhook url

- retrieve secret - kubectl get secret alertmanager-todo-dev-kube-prometheus-s-alertmanager-generated -n default -o yaml
- Decode the Configuration: echo
H4sIAAAAAAAA/7xSy07DMBC8+ytWPSKV8BCXSHxAv4BjtDUBZyU/wnqdqlLFtyMnUFKJCK5cZ2d3xjN2Pu3RtwZAKCc/UaccKBVt4SkYSUVpGVriiaSFTSzebwAk1TGBn+s4y1EDJRhtFTJdSsv+A+LAAHVDiQzpZLQk2i9AM/vsNnFPu3iwHvWJKExyn1N7lvygKwtPN7IM8JRSSbOs+dqdiTUFXr/MBheLnZSfPW2BUVxpN3ayxYyTSSsx2rgbI5uhPPhmAnlpYusoHK6xs0RsAeitLppe5rB76uz78UfbT5X+ormqCi5KuiX+VP+dd4fMvUaqJr12aaEhtMx8PGNGRNDbFn1zc6th9OYjAAD//xIx++qkAgAA | base64 --decode > alertmanager.yaml.gz
- Decompress the File: gzip -d alertmanager.yaml.gz
- Make the changes to the **alertmanager.yaml**
- Compress the file back- gzip alertmanager.yaml
- Recode it to base64- cat alertmanager.yaml.gz | base64 > encoded-alertmanager.txt
- replace the old encoded data for **alertmanager.yaml.gz** with the new base64 string from **encoded-alertmanager.txt**.
- Apply the changes to the Secret: kubectl apply -f secret.yaml
- Delete the Alertmanager Pods, basically force Kubernetes to force update- kubectl delete pods -l app.kubernetes.io/name=alertmanager -n default

```
...ank/Desktop/todo_app % zsh          ...Desktop/todo_app % zsh          ...kubectl + minikube dashboard + Done
Q: pred
Normal Created 17m kubelet      Created container config-reloader
Normal Started 17m kubelet      Started container config-reloader
sashank@ashanks-MBP todo_app % kubectl get secret alertmanager-todo-dev-kube-prometheus-s-alertmanager-generated -n default -o yaml
apiVersion: v1
data:
  .Alertmanager.yaml: H4sIAAAAAAAA/7xSy07DMBC8+ytWPSKV8BCXSHxAv4BjtDUBZyU/wnqdqlLFtyMnUFKJCK5cZ2d3xjN2Pu3RtwZAKCc/UaccKBVt4SkYSUVpGVriiaSFTSzebwAk1TGBn+s4y1EDJRhtFTJdSsv+A+LAAHVDiQzpZLQk2i9AM/vsNnFPu3iwHvWJKExyn1N7lvygKwtPN7IM8JRSSbOs+dqdiTUFXr/MBheLnZSfPW2BUVxpN3ayxYyTSSsx2rgbI5uhPPhmAnlpYusoHK6xs0RsAeitLppe5rB76uz78UfbT5X+ormqCi5KuiX+VP+dd4fMvUaqJr12aaEhtMx8PGNGRNDbFn1zc6th9OYjAAD//xIx++qkAgAA
kind: Secret
metadata:
  creationTimestamp: "2023-11-12T06:09:35Z"
  labels:
    managed-by: prometheus-operator
  name: alertmanager-todo-dev-kube-prometheus-s-alertmanager-generated
  namespace: default
  ownerReferences:
    - apiVersion: monitoring.coreos.com/v1
      blockOwnerDeletion: true
      controller: true
      kind: Alertmanager
    name: todo-dev-kube-prometheus-s-alertmanager
    uid: 7141-492-8a39-71c4e696dd2
  resourceVersion: "326244"
  uid: dde6b81-98a9-479e-b86d-38d2a9797ec
type: Opaque
sashank@ashanks-MBP todo_app % kubectl get secret alertmanager-todo-dev-kube-prometheus-s-alertmanager-generated -n default -o yaml > secret.yaml
sashank@ashanks-MBP todo_app % echo H4sIAAAAAAAA/7xSy07DMBC8+ytWPSKV8BCXSHxAv4BjtDUBZyU/wnqdqlLFtyMnUFKJCK5cZ2d3xjN2Pu3RtwZAKCc/UaccKBVt4SkYSUVpGVriiaSFTSzebwAk1TGBn+s4y1EDJRhtFTJdSsv+A+LAAHVDiQzpZLQk2i9AM/vsNnFPu3iwHvWJKExyn1N7lvygKwtPN7IM8JRSSbOs+dqdiTUFXr/MBheLnZSfPW2BUVxpN3ayxYyTSSsx2rgbI5uhPPhmAnlpYusoHK6xs0RsAeitLppe5rB76uz78UfbT5X+ormqCi5KuiX+VP+dd4fMvUaqJr12aaEhtMx8PGNGRNDbFn1zc6th9OYjAAD//xIx++qkAgAA --decode > alertmanager.yaml
sashank@ashanks-MBP todo_app % gzip alertmanager.yaml
sashank@ashanks-MBP todo_app % gzip alertmanager.yaml.gz
sashank@ashanks-MBP todo_app % cat alertmanager.yaml.gz | base64 > encoded-alertmanager.txt
sashank@ashanks-MBP todo_app % kubectl apply -f secret.yaml
Warning: resource secrets/alertmanager-todo-dev-kube-prometheus-s-alertmanager-generated is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required by kubectl apply. kubectl apply should only be used on resources created declaratively by either kubectl create --save-config or kubectl apply. The missing annotation will be patched automatically.
secret/alertmanager-todo-dev-kube-prometheus-s-alertmanager-generated configured
sashank@ashanks-MBP todo_app % kubectl delete pods -l app.kubernetes.io/name=alertmanager -n default
pod "alertmanager-todo-dev-kube-prometheus-s-alertmanager-*" deleted
sashank@ashanks-MBP todo_app % kubectl get pods -n default
NAME                      READY   STATUS    RESTARTS   AGE
alertmanager-todo-dev-kube-prometheus-s-alertmanager-0   2/2     Running   0          7s
my-flask-app-rc-c74pc     1/1     Running   0          34h
my-flask-app-rc-j3m7s     1/1     Running   0          34h
my-flask-service-rc-99m   1/1     Running   0          34h
my-flask-service-rc-lmhj  1/1     Running   0          34h
my-flask-app-rc-qnvk9    1/1     Running   0          34h
my-flask-app-rc-wj1kb    1/1     Running   0          34h
my-flask-service-deployment-94cc79495-4pmrz   1/1     Running   0          9h
my-flask-service-deployment-94cc79495-p99rt   1/1     Running   0          9h
my-flask-service-deployment-94cc79495-mm6m   1/1     Running   0          9h
my-flask-service-deployment-94cc79495-vcb02   1/1     Running   0          9h
my-flask-service-deployment-94cc79495-9dc4q   1/1     Running   0          9h
my-flask-service-deployment-94cc79495-zpd44   1/1     Running   0          9h
pod1-deployment-7cc4694ddd-9zctr   1/1     Running   0          35h
prometheus-todo-dev-prometheus-s-prometheus-0   2/2     Running   0          28m
todo-dev-kube-prometheus-fana-A0A77D99-7hrxf   1/1     Running   0          28m
todo-dev-kube-prometheus-s-operator-75884965-hnkf8  1/1     Running   0          28m
todo-dev-kube-state-metrics-f4d65945-mlc72   1/1     Running   0          28m
todo-dev-kube-state-metrics-f4d65945-mlc73   1/1     Running   0          28m
todo-dev-kube-state-node-exporter-9310n   1/1     Running   0          28m
sashank@ashanks-MBP todo_app % kubectl logs alertmanager-todo-dev-kube-prometheus-s-alertmanager-0
ts=2023-11-12T06:38:16.923Z caller=min.go:245 level=info msg="Starting Alertmanager" version="Version: 26.0, branch:HEAD, revision:d7ba4fe7-3927-49de-9b1c1b361c295d8d"
ts=2023-11-12T06:38:16.923Z caller=min.go:245 level=info msg="Using configuration file /etc/alertmanager/config_out/alertmanager.env.yaml"
ts=2023-11-12T06:38:16.973Z caller=coordinator.go:113 level=info component=configConfiguration msg="Loading configuration file /etc/alertmanager/config_out/alertmanager.env.yaml"
ts=2023-11-12T06:38:16.973Z caller=coordinator.go:126 level=info component=configConfiguration msg="Completed loading of configuration file" file=/etc/alertmanager/config_out/alertmanager.env.yaml
```

4. Create Alerting Rules: Define rules in prometheus.yml and perform kubectl apply -f prometheus.yml and perform kubectl get pods, kubectl get configmaps

```

[sashank@ashanks-MBP todo_app %] helm install todo-dev prometheus-community/kube-prometheus-stack
NAME: todo-dev
LAST DEPLOYED: Sun Nov 12 01:09:00 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
kube-prometheus-stack has been installed. Check its status by running:
  kubectl --namespace default get pods -l "release=todo-dev"

Visit https://github.com/prometheus-operator/kube-prometheus for instructions on how to create & configure Alertmanager and Prometheus instances using the Operator.
[sashank@ashanks-MBP todo_app %] kubectl apply -f prometheus.yml
prometheusoperator.monitoring.coreos.com/probe-alerts created
[sashank@ashanks-MBP todo_app %] kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
alertmanager-todo-dev-kube-prometheus-s-alertmanager-0   2/2    Running   0          92s
my-flask-app-rc-c74qc              1/1    Running   0          33h
my-flask-app-rc-j5n7s              1/1    Running   0          33h
my-flask-app-rc-jcm                1/1    Running   0          33h
my-flask-app-rc-lmpj1              1/1    Running   0          33h
my-flask-app-rc-qnvk9              1/1    Running   0          33h
my-flask-app-rc-wjks              1/1    Running   0          33h
my-flask-service-deployment-94cc79695-1gmrk               1/1    Running   0          9h
my-flask-service-deployment-94cc79695-pe9st               1/1    Running   0          9h
my-flask-service-deployment-94cc79695-qm6m                1/1    Running   0          9h
my-flask-service-deployment-94cc79695-vcbdz               1/1    Running   0          9h
my-flask-service-deployment-94cc79695-zfdq                1/1    Running   0          9h
my-flask-service-deployment-94cc79695-zpd4                1/1    Running   0          9h
pod-disruption-budget-7cc694add-5zcrt                  1/1    Running   0          34h
prometheus-todo-dev-kube-prometheus-s-prometheus-0       2/2    Running   0          92s
todo-dev-grafana                   1/1    Running   0          97s
todo-dev-kube-prometheus-s-operator-67588a495-hvxk8      1/1    Running   0          97s
todo-dev-kube-state-metrics-f4ab59845-vc1c72             1/1    Running   0          97s
todo-dev-prometheus-node-exporter-8qyjq                 1/1    Running   0          97s
todo-dev-prometheus-node-exporter-8qyjq-190               1/1    Running   0          97s
[sashank@ashanks-MBP todo_app %] kubectl get configmaps
NAME                           DATA   AGE
kube-prometheus-svc             1     35h
prometheus-todo-dev-kube-prometheus-s-prometheus-rulefiles-0 29    18m
todo-dev-grafana                1     10m
todo-dev-grafana-config-dashboards 1     10m
todo-dev-kube-prometheus-s-alertmanager-overview           1     10m
todo-dev-kube-prometheus-s-cluster           1     10m
todo-dev-kube-prometheus-s-cluster-total         1     10m
todo-dev-kube-prometheus-s-controller-manager        1     10m
todo-dev-kube-prometheus-s-coredns            1     10m
todo-dev-kube-prometheus-s-grafana-datasource       1     10m
todo-dev-kube-prometheus-s-grafana-overview        1     10m
todo-dev-kube-prometheus-s-k8s-coredns           1     10m
todo-dev-kube-prometheus-s-k8s-node-cluster         1     10m
todo-dev-kube-prometheus-s-k8s-resources-mycluster 1     10m
todo-dev-kube-prometheus-s-k8s-resources-namespace 1     10m
todo-dev-kube-prometheus-s-k8s-resources-node       1     10m
todo-dev-kube-prometheus-s-k8s-resources-pod        1     10m
todo-dev-kube-prometheus-s-k8s-resources-workload 1     10m
todo-dev-kube-prometheus-s-k8s-resources-workload-namespace 1     10m
todo-dev-kube-prometheus-s-kubelet                   1     10m
todo-dev-kube-prometheus-s-kubernetes-space-by-pod 1     10m
todo-dev-kube-prometheus-s-kubernetes-space-by-workload 1     10m
todo-dev-kube-prometheus-s-node-cluster-rsrc-use   1     10m
todo-dev-kube-prometheus-s-node-rsrc-use           1     10m
todo-dev-kube-prometheus-s-nodes-darwin            1     10m
todo-dev-kube-prometheus-s-persistentvolumeusage   1     10m
todo-dev-kube-prometheus-s-pod-total              1     10m
todo-dev-kube-prometheus-s-prometheus             1     10m
todo-dev-kube-prometheus-s-proxy                  1     10m
todo-dev-kube-prometheus-s-scheduler             1     10m
todo-dev-kube-prometheus-s-workload-total        1     10m
[sashank@ashanks-MBP todo_app %] kubectl describe configmap todo-dev-kube-prometheus-s-alertmanager-overview

```

5. Test the alerting system by deleting one of the pods or hitting the test curl endpoint of alertmanager and you should receive an alert in the slack channel

```

[sashank@ashanks-MBP todo_app %] kubectl delete pod my-flask-service-deployment-94cc79695-vcbdz
pod "my-flask-service-deployment-94cc79695-vcbdz" deleted
[sashank@ashanks-MBP todo_app %] kubectl get svc
NAME                           TYPE      CLUSTER-IP      EXTERNAL-IP                                     PORT(S)          AGE
alertmanager-operated          ClusterIP  None           <none>
kubernetes                      ClusterIP  10.100.0.1    <none>
my-flask-service                LoadBalancer 10.100.224.232  a94dba753295c48fea738acc5cc7b575-293773832.us-west-2.elb.amazonaws.com  9093/TCP,9094/TCP,9094/UDP  42m
pod1-service                    ClusterIP  10.100.135.68 <none>
prometheus-operated            ClusterIP  None           <none>
todo-dev-grafana                ClusterIP  10.100.113.145 <none>
todo-dev-kube-prometheus-s-alertmanager ClusterIP  10.100.197.118 <none>
todo-dev-kube-prometheus-s-operator  ClusterIP  10.100.240.137 <none>
todo-dev-kube-prometheus-s-prometheus ClusterIP  10.100.94.158 <none>
todo-dev-kube-state-metrics      ClusterIP  10.100.79.0   <none>
todo-dev-prometheus-node-exporter ClusterIP  10.100.155.189 <none>
[sashank@ashanks-MBP todo_app %] curl -XPOST -d'{
  "labels": {
    "alertname": "TestAlert",
    "severity": "critical"
  },
  "annotations": {
    "summary": "This is a test alert"
  }
}' http://10.100.197.118:9093/api/v1/alerts

```