

Team Members:

1. Reet Nandy - rn2528
2. Rohan Chopra - rc4920

Part 1: Creating an Application

- Download To-Do web application built using Flask and MongoDB.
 - Modify the app.py file to change the host to 0.0.0.0, and add Werkzeug==2.2.2 to the requirements.txt file.

The screenshot shows a developer's workspace with several tabs open across different windows:

- Cloud Computing** tab: A browser window showing a todo reminder application.
- EXPLORER** tab: A sidebar showing project structure and files like `app.py`, `index.html`, and `requirements.txt`.
- TERMINAL** tab: A terminal window showing command-line output related to the application's deployment.
- OUTPUT** tab: A log or output window showing deployment logs.
- PREVIEW** tab: A preview window showing the application's interface.

The `app.py` file content is as follows:

```
Assignment >> app.py >>
1 from flask import Flask, render_template, request, redirect, url_for
2 from pymongo import MongoClient
3 from bson.objectid import ObjectId
4 from bson.json_util import dumps
5 import os
6
7 mongodb_host = os.environ.get('MONGO_HOST', 'localhost')
8 mongodb_port = int(os.environ.get('MONGO_PORT', '27017'))
9 client = MongoClient(mongodb_host, mongodb_port)
10 db = client.camp2016
11 todos = db.todos
12
13 app = Flask(__name__)
14 title = "TODO with Flask"
15 heading = "ToDo Reminder"
16
17 def redirect_url():
18     return request.args.get('next') or request.referrer or url_for('index')
19
20 @app.route('/list')
21 def lists():
22     todos_l = todos.find()
23     al = "active"
24
25     return render_template('index.html', al=al, todos=todos_l, title=title, heading=heading)
26
27 if __name__ == '__main__':
28     app.run(host='0.0.0.0', port=5000)
```

The terminal window shows deployment logs:

```
File "/Users/techerzetti/Desktop/Code/cloudComputing/cloudcomputing/lib/python3.9/site-packages/pymongo/cursor.py", line 1139, in _refresh
    self.__send_message("TAILABLEAFTER")
File "/Users/techerzetti/Desktop/Code/cloudComputing/cloudcomputing/lib/python3.9/site-packages/pymongo/mongo_client.py", line 1712, in __ensure_session
    self._start_session()
File "/Users/techerzetti/Desktop/Code/cloudComputing/cloudcomputing/lib/python3.9/site-packages/pymongo/mongo_client.py", line 1637, in _start_session
    self._select_server(selector)
File "/Users/techerzetti/Desktop/Code/cloudComputing/cloudcomputing/lib/python3.9/site-packages/pymongo/topology.py", line 538, in _check_implicit_session_support
    self._select_servers_loop(
File "/Users/techerzetti/Desktop/Code/cloudComputing/cloudcomputing/lib/python3.9/site-packages/pymongo/topology.py", line 238, in _select_servers_loop
    raise ServerSelectionTimeoutError(
pymongo.errors.ServerSelectionTimeoutError: localhost:27017: [Errno 61] Connection refused, Timeout: 30s, Topology Description: <TopologyDescription id: 65f8932d85eedaa439b65aae, topology_type: Unknown, servers: [

The browser window shows the application's interface with a list of todos.


```

Part 2: Containerizing the Application with Docker

- Install Docker using Homebrew (brew).
- Utilize two containers: one for the Flask application and another for MongoDB.
- Craft a Dockerfile for the Flask application.
- **docker build -t todo-app .**
- Draft a docker-compose file to facilitate local testing.
- Proceed to push the Docker image to Docker Hub using **docker-compose up**.
- Log in to Docker Hub using **docker login**.
- Tag the Docker image as follows: **docker tag todo_app techpertz/todo_app:latest**.
- Finally, push the Docker image to Docker Hub: **docker push techpertz/todo-app**.

```
version: '3'
services:
  app:
    image: todo_app
    ports:
      - "5000:5000"
    depends_on:
      - mongo
  mongo:
    image: mongo
    volumes:
      - mongo-data:/data/db
```

```
version: '3'
services:
  app:
    image: todo_app
    ports:
      - "5000:5000"
    depends_on:
      - mongo
  mongo:
    image: mongo
    volumes:
      - mongo-data:
```

```
from flask import Flask, request
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

The screenshot shows a browser window with the URL `127.0.0.1:5000/about`. The title bar says "ToDo Reminder". Below the title bar is a dark navigation bar with three tabs: "ALL", "Uncompleted", and "Completed". On the far right of the navigation bar is an "About" link. The main content area is completely empty, showing only a few horizontal lines.

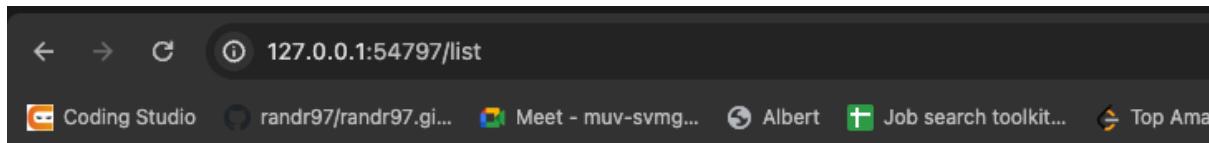
The screenshot shows a browser window on the Docker Hub website at the URL `hub.docker.com/repository/docker/techpertz/todo-app/general`. The page title is "techpertz/todo-app". The main content area includes sections for "Docker commands", "Tags", "Automated Builds", and "Repository overview". The "Tags" section shows one tag: "latest". The "Automated Builds" section indicates that manual pushing is required. The "Repository overview" section provides a brief description of the image's purpose.

Step 3: Deploying the Application on Minikube

- Install Minikube.
- Craft a YAML file for Kubernetes deployment.
- Start Minikube using the command: **minikube start**.
- Log in to Docker using: **docker login**.
- Establish pods for both the Flask application and MongoDB.
- Create a Kubernetes deployment for the application.
- Proceed to create and expose the Kubernetes deployment using: **kubectl apply -f minikube.yaml**.
- Verify the status of the pods with: **kubectl get pods**.
- Test the application functionality on Minikube by running: **minikube service my-flask-service**.
- Now delete the pod and apply again to check if data still persists
- **kubectl delete pod pod1**
- **kubectl apply -f minikube.yaml**

The screenshot shows a terminal window with several tabs: PROBLEMS, PORTS, COMMENTS, OUTPUT, TERMINAL (which is selected), and DEBUG CONSOLE. The terminal output is as follows:

```
51 ---  
PROBLEMS 4 PORTS COMMENTS OUTPUT TERMINAL DEBUG CONSOLE  
● (cloudcomputing) (base) techpertz@Reets-MBP Cloud Computing % minikube start  
minikube v1.32.0 on Darwin 14.3.1  
Using the docker driver based on existing profile  
Starting control plane node minikube in cluster minikube  
Pulling base image ...  
Updating the running docker "minikube" container ...  
Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...  
Verifying Kubernetes components...  
  Using image gcr.io/k8s-minikube/storage-provisioner:v5  
  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  
  
* Enabled addons: storage-provisioner, default-storageclass, dashboard  
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default  
● (cloudcomputing) (base) techpertz@Reets-MBP Cloud Computing % kubectl apply -f minikube.yaml  
error: the path "minikube.yaml" does not exist  
● (cloudcomputing) (base) techpertz@Reets-MBP Cloud Computing % ls  
Assignment Quiz 1 Weekly Tasks cloudcomputing  
● (cloudcomputing) (base) techpertz@Reets-MBP Cloud Computing % cd Assignment/2  
● (cloudcomputing) (base) techpertz@Reets-MBP 2 % ls  
Archive app.py flask-rc.yaml requirements.txt static  
Dockerfile docker-compose.yml minikube.yaml sashank templates  
● (cloudcomputing) (base) techpertz@Reets-MBP 2 % kubectl apply -f minikube.yaml  
persistentvolumeclaim/mongo-persistent-storage unchanged  
service/mongo-service unchanged  
deployment.apps/mongo unchanged  
deployment.apps/todo-app configured  
service/todo-app-service unchanged  
● (cloudcomputing) (base) techpertz@Reets-MBP 2 % kubectl get pods  
NAME READY STATUS RESTARTS AGE  
mongo-64b5667964-j4n7z 1/1 Running 1 (100s ago) 3h36m  
todo-app-56fc88ddd6-hqchc 0/1 ImagePullBackOff 0 24s  
todo-app-6fcacfb6f-p4jcd 1/1 Running 1 (99s ago) 151m  
todo-app-6fcacfb6f-sj9bt 1/1 Running 1 (99s ago) 3h13m  
● (cloudcomputing) (base) techpertz@Reets-MBP 2 % minikube service todo-app-service  
NAMEPORT/PORT URL  
default todo-app-service 80 http://192.168.49.2:32021  
  
Starting tunnel for service todo-app-service.  
NAMEPORT/PORT URL  
default todo-app-service 80 http://127.0.0.1:54797  
  
Opening service default/todo-app-service in default browser...  
Because you are using a Docker driver on darwin, the terminal needs to be open to run it.
```



ToDo Reminder

ALL

Uncompleted

Completed

Search Reference

To-Do LIST :

Status	Task Name
X	Reet test

Add a Task

Taskname

Enter Description here...

mm/dd/yyyy

Priority

```
(cloudcomputing) (base) techpertz@Reets-MBP 2 % kubectl delete pod todo-app-56fc88ddd6-hqchc
pod "todo-app-56fc88ddd6-hqchc" deleted
(cloudcomputing) (base) techpertz@Reets-MBP 2 % kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
mongo-64b5667964-j4n7z   1/1     Running   1 (5m21s ago)   3h40m
todo-app-56fc88ddd6-sbxhc  0/1     ErrImagePull   0           8s
todo-app-6fcdfcfbd6f-p4jcd  1/1     Running   1 (5m20s ago)   155m
todo-app-6fcdfcfbd6f-sj9bt  1/1     Running   1 (5m20s ago)   3h17m
(cloudcomputing) (base) techpertz@Reets-MBP 2 %
```

ToDo Reminder

ALL Uncompleted Completed

Search Reference: Unique ID Search

Status	Task Name	Description Name	Date	Priority	Remove	Modify
X	Reet test	Reet Test	2024-03-05	Medium !!		

Add a Task

Taskname

Enter Description here...

mm/dd/yyyy

Priority

Create

kubernetes default Search

Workloads

Workload Status

Deployments Pods Replica Sets

Deployments

Name	Images	Labels	Pods	Created
mongo	mongo	-	1 / 1	3 hours ago
todo-app	637423495437.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v8	-	2 / 2	3 hours ago

Pods

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
todo-app-5fc88ddd-stbhc	637423495437.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v3	app: todo-app pod-template-hash: 5fc88dd	minikube	ImagePullBackOff	-	-	-	a minute ago
todo-app-6fdcfbd6-p4jcd	techpertz/todo-app:v2	app: todo-app pod-template-hash: 6fdcfbd6	minikube	Running	1	-	-	2 hours ago
todo-app-6fdcfbd6-sj9bt	techpertz/todo-app:v2	app: todo-app pod-template-hash: 6fdcfbd6	minikube	Running	1	-	-	3 hours ago
mongo-64b567964-j4n7z	mongo	app: mongo pod-template-hash: 64b567964	minikube	Running	1	-	-	3 hours ago

Replica Sets

Name	Images	Labels	Pods	Created
todo-app-5fc88ddd	637423495437.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v8	app: todo-app pod-template-hash: 5fc88ddd	0 / 1	5 minutes ago

Kubernetes Dashboard - Services

Services

Name	Labels	Type	Cluster IP	Internal Endpoints	External Endpoints	Created
mongo-service	-	ClusterIP	10.102.178.230	mongo-service:27017 TCP mongo-service:0 TCP	-	3 hours ago
todo-app-service	-	LoadBalancer	10.104.233.2	todo-app-service:80 TCP todo-app-service:32021 TCP	-	3 hours ago
kubernetes	component: apiserver provider: kubernetes	ClusterIP	10.96.0.1	kubernetes:443 TCP kubernetes:0 TCP	-	3 hours ago

Workloads

- Cron Jobs
- Daemon Sets
- Deployments
- Jobs
- Pods
- Replica Sets
- Replication Controllers
- Stateful Sets

Service

- Ingresses
- Ingress Classes
- Services

Config and Storage

- Config Maps
- Persistent Volume Claims
- Secrets
- Storage Classes

Cluster

- Cluster Role Bindings
- Cluster Roles

Kubernetes Dashboard - Persistent Volumes

Persistent Volumes

Name	Capacity	Access Modes	Reclaim Policy	Status	Claim	Storage Class	Reason	Created
pvc-49164f78-73b9-4530-a7a3-0de05cbe31e3	storage: 1Gi	ReadWriteOnce	Delete	Bound	default/mongo-persistent-storage	standard	-	3 hours ago

Workloads

- Cron Jobs
- Daemon Sets
- Deployments
- Jobs
- Pods
- Replica Sets
- Replication Controllers
- Stateful Sets

Service

- Ingresses
- Ingress Classes
- Services

Config and Storage

- Config Maps
- Persistent Volume Claims
- Secrets
- Storage Classes

Cluster

- Cluster Role Bindings
- Cluster Roles
- Events
- Namespaces
- Network Policies
- Nodes
- Persistent Volumes
- Role Bindings
- Roles
- Service Accounts

Custom Resource Definitions

- Settings
- About

Step 4:

- Create 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**

```
base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)z12 (16m 15.55s)
eksctl create cluster --name todo-cluster \
    --version 1.23 \
    --region us-west-2 \
    --nodegroup-name amazon-nodes \
    --node-type t3.medium \
    --nodes-min 1 \
    --nodes-max 4 \
    --managed

[2024-03-18 18:10:22] [i] nodegroup "amazon-nodes" will use -- [AMAZONLINUX/1.23]
[2024-03-18 18:10:22] [i] using Kubernetes version 1.23
[2024-03-18 18:10:22] [i] creating EKS cluster "todo-cluster" in "us-west-2" region with managed nodes
[2024-03-18 18:10:22] [i] it will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
[2024-03-18 18:10:22] [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-west-2 --cluster=todo-cluster'
[2024-03-18 18:10:22] [i] Kubernetes API endpoint access will use default of [publicAccess=true, privateAccess=false] for cluster "todo-cluster" in "us-west-2"
[2024-03-18 18:10:22] [i] CloudWatch logging will not be enabled for cluster "todo-cluster" in "us-west-2"
[2024-03-18 18:10:22] [i] 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'
[2024-03-18 18:10:22] [i]

2 sequential tasks: { create cluster control plane "todo-cluster",
  2 sequential sub-tasks: {
    wait for control plane to become ready,
    create managed nodegroup "amazon-nodes",
  }
}

[2024-03-18 18:10:22] [i] building cluster stack "eksctl-todo-cluster-cluster"
[2024-03-18 18:10:23] [i] deploying stack "eksctl-todo-cluster-cluster"
[2024-03-18 18:10:23] [i] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
[2024-03-18 18:11:23] [i] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
[2024-03-18 18:12:24] [i] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
[2024-03-18 18:13:25] [i] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
[2024-03-18 18:14:25] [i] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
[2024-03-18 18:15:26] [i] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
[2024-03-18 18:16:26] [i] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
[2024-03-18 18:17:27] [i] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
[2024-03-18 18:18:27] [i] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
[2024-03-18 18:19:28] [i] waiting for CloudFormation stack "eksctl-todo-cluster-cluster"
[2024-03-18 18:20:28] [i] waiting for CloudFormation stack "eksctl-todo-cluster-nodegroup"
[2024-03-18 18:22:34] [i] building managed nodegroup stack "eksctl-todo-cluster-nodegroup-amazon-nodes"
[2024-03-18 18:22:35] [i] deploying stack "eksctl-todo-cluster-nodegroup-amazon-nodes"
[2024-03-18 18:23:35] [i] waiting for CloudFormation stack "eksctl-todo-cluster-nodegroup-amazon-nodes"
[2024-03-18 18:23:35] [i] waiting for CloudFormation stack "eksctl-todo-cluster-nodegroup-amazon-nodes"
[2024-03-18 18:24:42] [i] waiting for CloudFormation stack "eksctl-todo-cluster-nodegroup-amazon-nodes"
[2024-03-18 18:24:42] [i] waiting for CloudFormation stack "eksctl-todo-cluster-nodegroup-amazon-nodes"
[2024-03-18 18:26:33] [i] waiting for CloudFormation stack "eksctl-todo-cluster-nodegroup-amazon-nodes"
[2024-03-18 18:26:33] [i] waiting for the control plane to become ready
[2024-03-18 18:26:33] [x] saved kubeconfig as "/Users/techpertz/.kube/config"
[2024-03-18 18:26:33] [x] not trusted
[2024-03-18 18:26:33] [x] all EKS cluster resources for "todo-cluster" have been created
[2024-03-18 18:26:34] [i] nodegroup "amazon-nodes" has 3 node(s)
[2024-03-18 18:26:34] [i] node "ip-192-168-30-51.us-west-2.compute.internal" is ready
[2024-03-18 18:26:34] [i] node "ip-192-168-33-199.us-west-2.compute.internal" is ready
[2024-03-18 18:26:34] [i] node "ip-192-168-33-200.us-west-2.compute.internal" is ready
[2024-03-18 18:26:34] [i] waiting for at least 1 node(s) to become ready in "amazon-nodes"
[2024-03-18 18:26:35] [i] nodegroup "amazon-nodes" has 3 node(s)
[2024-03-18 18:26:35] [i] node "ip-192-168-30-51.us-west-2.compute.internal" is ready
[2024-03-18 18:26:35] [i] node "ip-192-168-33-199.us-west-2.compute.internal" is ready
[2024-03-18 18:26:35] [i] node "ip-192-168-33-200.us-west-2.compute.internal" is ready
[2024-03-18 18:26:36] [i] Kubectl command should work with "/Users/techpertz/.kube/config", try 'kubectl get nodes'
[2024-03-18 18:26:36] [x] EKS cluster "todo-cluster" in "us-west-2" region is ready
```

Instances | EC2 | us-west-2 todo-cluster | Clusters | Eks | AWS Service Role for Amazon EKS | Amazon EKS Service Role Policy | +

Coding Studio randy97/randy97.g... Meet - muv-svmp... Albert Job search toolkit... Top Amazon Ques... Matching Fuzzy Te... All Bookmarks

aws Services Search [Option+S] Upgrade version

Amazon Elastic Kubernetes Service

Clusters New

Amazon EKS Anywhere

Enterprise Subscriptions New

Related services

Amazon ECR

AWS Batch

Documentation

Submit feedback

A new Kubernetes version is available for this cluster. Upgrade version

EKS > Clusters > todo-cluster

todo-cluster

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. [Learn more](#) [Update now](#)

▼ Cluster info Info

Status	Kubernetes version	Support type	Provider
Active	Info 1.23	Extended support until October 11, 2024	EKS

Overview Resources Compute Networking Add-ons Access Observability Upgrade insights Update history Tags

Details

API server endpoint	OpenID Connect provider URL	Created
https://ED999D1598770E80526295218B98C87B.us-west-2.eks.amazonaws.com	https://oidc.eks.us-west-2.amazonaws.com/id/ED999D1598770E80526295218B98C87B	4 hours ago
Certificate authority	Cluster IAM role ARN	Cluster ARN
LS0LhLS1CRUdjTlBDRVJUSUZJQ0FURS0L50Ick1JSUVMvaKNDOQWWh20FSUJbZ0CQURtKna3Fa21H0QxwqKFRC0ZBREFWTvJnd0VRWURWUVFERXdwcmRXSmwKY201	arn:aws:iam::637423495453:role/eksctl-todo-cluster-ServiceRole-7g0RdhNews7	arn:aws:eks:us-west-2:637423495453:cluster/todo-cluster
Platform version	Info	ebs.21

Health issues (0)

Issue type	Description	Affected resources
------------	-------------	--------------------

- #### - Configure kubectl for EKS

- Create an ECR Image **aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-stdin**
637423495437.dkr.ecr.us-west-2.amazonaws.com
- Build the Docker Image
- Create ECR Repository **aws ecr create-repository --repository-name todo-flask-app --region us-west-2**
- Tag the Docker Image for ECR **docker tag todo-flask-app:v2**
637423495437.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v2
- Push the Image to Amazon ECR **docker push**
637423495437.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v2

```

. .Computing/Assignment2 + 
base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)z12 (1m 6.48s)
docker tag todo-flask-app:v2 637423495437.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v2

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)z12 (1m 6.48s)
docker push 637423495437.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v2

The push refers to repository [637423495437.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app]
abe32228de91: Pushed
637423495437: Pushed
6b2a534e5569: Pushed
3482d4cd60de: Pushed
370c9e78e3ea: Pushed
a74bee0a4845: Pushed
c8f253ae5f88: Pushed
a483d0aa8a369: Pushed
v2: digest: sha256:c146d724aa37269867b2385a4c6142e9f9b376c8802305eafb2214c6b7d02a4f size: 1999

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

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)z12 (1.20s)
aws ecr list-images --repository-name todo-flask-app --region us-west-2

{
  "imageIds": [
    {
      "imageDigest": "sha256:c146d724aa37269867b2385a4c6142e9f9b376c8802305eafb2214c6b7d02a4f",
      "imageTag": "v2"
    }
  ]
}

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)z12 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (4.61s)
kubectl apply -f minikube.yaml

persistentvolumeclaim/mongo-persistent-storage created
service/mongo-service created
deployment.apps/mongo created
deployment.apps/todo-app created
service/todo-app-service created

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)z12 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (1.20s)
kubectl get pods -n kube-system

NAME          READY   STATUS    RESTARTS   AGE
aws-node-2ggzd  2/2     Running   0          38m
aws-node-4rt   2/2     Running   0          38m
aws-node-vndka  2/2     Running   0          38m
coredns-7586478bb5-gdfj12  1/1     Running   0          44m
coredns-7586478bb5-rt5pf  1/1     Running   0          44m
kube-proxy-4slk2  1/1     Running   0          38m
kube-proxy-czqmm  1/1     Running   0          38m
kube-proxy-ktkpp  1/1     Running   0          38m

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)z13

```

- Set up iam roles for EKS to interact with Elas6c Block Store (EBS) and make sure you finally get the AmazonEKS_EBS_CSI_Driver

```

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)z12 (3.757s)
brew install helm

#####
## Downloading https://formulae.brew.sh/api/cask.jws.json
#####
Warning: helm 3.14.3 is already installed and up-to-date.
To reinstall 3.14.3, run:
  brew reinstall helm

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)z12 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (0.873s)
kubectl config current-context

arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)z12 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (1.26s)
helm repo add aws-ebs-csi-driver https://kubernetes-sigs.github.io/aws-ebs-csi-driver/
"aws-ebs-csi-driver" has been added to your repositories

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)z12 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (2.134s)
helm repo update

Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "aws-ebs-csi-driver" chart repository
...Successfully got an update from the "prometheus-community" chart repository
...Successfully got an update from the "stable" chart repository
Update Complete. *Happy Helm-ing*

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)z12 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (7.045s)
helm install aws-ebs-csi-driver aws-ebs-csi-driver/aws-ebs-csi-driver --namespace kube-system --set enableVolumeScheduling=true --set enableVolumeResizing=true --set enableVolumeSnapshot=true

NAME: aws-ebs-csi-driver
LAST DEPLOYED: Mon Mar 18 19:04:53 2024
NAMESPACE: kube-system
STATUS: deployed
REVISION: 1
NOTES:
To verify that aws-ebs-csi-driver has started, run:
  kubectl get pod -n kube-system -l "app.kubernetes.io/name=aws-ebs-csi-driver,app.kubernetes.io/instance=aws-ebs-csi-driver"
NOTE: The [CSI Snapshotter](https://github.com/kubernetes-csi/external-snapshotter) controller and CRDs will no longer be installed as part of this chart and moving forward will be a prerequisite of using the snap shotting functionality.
WARNING: Upgrading the EBS CSI Driver Helm chart with --reuse-values will no longer be supported in a future release. For more information, see https://github.com/kubernetes-sigs/aws-ebs-csi-driver/issues/1864

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)z12 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (1.319s)
helm ls -n kube-system

NAME          NAMESPACE   REVISION  UPDATED           STATUS        CHART
aws-ebs-csi-driver  kube-system  1          2024-03-18 19:04:53.226559 -0400 EDT  deployed    aws-ebs-csi-driver-2.28.1      1.28.8

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)z13

```

```

..Computing/Assignment/2

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±12 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (1.57s)
kubectl get pods -n kube-system | grep ebs-csi
ebs-csi-controller-7f9cd877b7-lkfb4 5/5 Running 0 6m51s
ebs-csi-controller-7f9cd877b7-tvgh 5/5 Running 0 6m51s
ebs-csi-node-d4qvf 3/3 Running 0 6m51s
ebs-csi-node-dpr7c 3/3 Running 0 6m51s
ebs-csi-node-sd6dq 3/3 Running 0 6m51s

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±12 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (1.07s)
kubectl get csidriver
NAME ATTACHREQUIRED PODINFOONMOUNT STORAGECAPACITY TOKENREQUESTS REQUIRESRERUBLISH MODES AGE
ebs.csi.aws.com true false false <unset> false Persistent 7m1s
efs.csi.aws.com false false false <unset> false Persistent 55m

```

The screenshot shows the AWS IAM Roles page. On the left, the navigation menu includes 'Identity and Access Management (IAM)', 'Dashboard', 'Access management' (with 'User groups', 'Users', 'Roles' selected), 'Policies', 'Identity providers', and 'Account settings'. Below these are 'Access reports' (with 'Access Analyzer', 'External access', 'Unused access', 'Analyzer settings', 'Credential report', 'Organization activity', and 'Service control policies (SCPs)'). Under 'Related consoles', there are links to 'IAM Identity Center' and 'AWS Organizations'. At the bottom, there are links for 'CloudShell' and 'Feedback'.

The main content area displays the details for the 'AWSServiceRoleForAmazonEKS' role. The title is 'AWSServiceRoleForAmazonEKS' with a 'Info' link. A description states: 'Allows Amazon EKS to call AWS services on your behalf.' To the right is a 'Delete' button. Below this is a 'Summary' section with fields: 'Creation date' (March 16, 2024, 17:28 (UTC-04:00)), 'ARN' (arn:aws:iam::637423495437:role/aws-service-role/eks.amazonaws.com/AWSServiceRoleForAmazonEKS), 'Last activity' (31 minutes ago), and 'Maximum session duration' (1 hour). Below the summary is a 'Permissions' tab, which is currently selected, showing a table for 'Permissions policies (1)'. The table has columns for 'Policy name' (AmazonEKSServiceRoleP...), 'Type' (AWS managed), and 'Attached entities' (1). There is also a 'Filter by Type' dropdown set to 'All types'.

- Create a Kubernetes deployment in EKS **kubectl apply -f minikube.yaml**

- Check the pods and service **kubectl get pods , kubectl get services, kubectl get pvc**

```

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±12 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (2.663s)
kubectl apply -f minikube.yaml

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±12 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (4.438s)
kubectl get deployments
kubectl get svc
kubectl get pods
kubectl get pvc

NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mongo     1/1     2/2         1           39m
todo-app  2/2     2/2         2           39m
NAME      TYPE        CLUSTER-IP      EXTERNAL-IP
kubernetes  ClusterIP    <none>        <none>
mongo-service ClusterIP    10.100.48.80   <none>
todo-app-service LoadBalancer  10.100.169.180  a481d2ce94e6e4a328e653cf618d8900-429554333.us-west-2.elb.amazonaws.com  80:30282/TCP  39m
NAME      READY   STATUS    RESTARTS   AGE
mongo-56dbcd9457-clnvnc  1/1     Running   0          39m
todo-app-6847448b7d-bfpocz 1/1     Running   0          39m
todo-app-6847448b7d-x5qj5  1/1     Running   0          39m
NAME      STATUS    VOLUME
mongo-persistent-storage  Bound    pvc-c5b29ebc-9540-428e-87ad-37c1eac70294  1Gi      RWO
                                     ACCESS MODES  STORAGECLASS  AGE
                                         gp2          39m

```

- Testing Application

The screenshot shows a web-based application titled "ToDo Reminder". The interface includes a navigation bar with tabs for "ALL", "Uncompleted", and "Completed". Below the navigation is a search bar with "Search Reference:" and "Unique ID" dropdowns, a "Search Task" button, and a "Search" button. A table displays a list of tasks:

Status	Task Name	Description Name	Date	Priority	Remove	Modify
X	Reet 1	Reet 1	2024-03-18	Low !		

Below the table is a form for adding a new task:

Add a Task

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

- Deleting pod and checking if data is still persistent in Incognito window.

```

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±12 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (1.996s)
kubectl delete pod mongo-56dbcd9457-clnvnc

pod "mongo-56dbcd9457-clnvnc" deleted

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±12 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (7m 40.97s)
kubectl get pods -w

NAME      READY   STATUS    RESTARTS   AGE
mongo-56dbcd9457-2hpcs  1/1     Running   0          8s
todo-app-6847448b7d-bfpocz 1/1     Running   0          44m
todo-app-6847448b7d-x5qj5  1/1     Running   0          44m

```

The screenshot shows the same "ToDo Reminder" application in an Incognito window. The interface and task list are identical to the previous screenshot, confirming that the data remains persistent even after deleting a pod.

Step 5: Replication Controller Feature

- Create a YAML file for the replication controller.
- Update the desired number of replicas as 3.
- Apply the configuration with: **kubectl apply -f flask-rc.yaml**.
- Validate the functionality of the replication controller through **kubectl get pods -l app=flask-rc-app**:

```
base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (2.224s)
kubectl apply -f flask-rc.yaml

replicationcontroller/flask-app-rc created

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (1.292s)
kubectl get pods -l app=flask-rc-app

NAME      READY   STATUS    RESTARTS   AGE
flask-app-rc-f4286  1/1     Running   0          9s
flask-app-rc-m2j4t  1/1     Running   0          9s
flask-app-rc-zth6k  1/1     Running   0          9s
```

- Delete a pod using **kubectl delete pod -l app=flask-rc-app**.
- Check the pods again with **kubectl get pods -l app=flask-rc-app** and observe the restarting pods.
- Lastly, confirm the replication controller status with **kubectl get rc flask-app-rc**

```
base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (3.092s)
kubectl delete pod -l app=flask-rc-app

pod "flask-app-rc-f4286" deleted
pod "flask-app-rc-m2j4t" deleted
pod "flask-app-rc-zth6k" deleted

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (1.248s)
kubectl get pods -l app=flask-rc-app

NAME      READY   STATUS    RESTARTS   AGE
flask-app-rc-l9k6b  1/1     Running   0          10s
flask-app-rc-snr7d  1/1     Running   0          10s
flask-app-rc-tnrz9  1/1     Running   0          10s

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (1.283s)
kubectl get rc flask-app-rc

NAME      DESIRED   CURRENT   READY   AGE
flask-app-rc  3         3         3       3m32s
```

Part 6: Rolling Update Strategy

- Configure the deployment to facilitate rolling updates.
- Proceed to build a new Docker image for the modified project and push it to the Amazon ECR repository.
- Update the deployment.yml file to incorporate the URL of this new Docker image.
- Execute the following Docker commands:
 - **docker build -t 637423495437.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v3 .**
 - **docker push 637423495437.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v3**
- Next, implement and execute the rolling update strategy.
- Apply the changes to the deployment by running **kubectl apply -f minikube.yaml**.

```

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13 (7.63s)
docker build -t 637423495437.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v3 .

[+] Building 6.4s [1/1] FINISHED
--> [internal] load build definition from Dockerfile
--> [internal] transfer dockerfile: 557B
--> [internal] load metadata for docker.io/library/python:3.8-slim
--> [auth] docker.io/library/python:pull token for registry-1.docker.io
--> [internal] load dockerignore
--> [internal] transfer dockerignore
--> [internal] load context
--> [internal] transfer context: 11.62kB
--> [internal] transfer requirements.txt: 0B
--> [internal] transfer app: 0B
--> [internal] transfer requirements.txt: 0B
--> [internal] transfer layers
--> [internal] writing manifest sha256:7022689955d3b88a2bfdb38e904b742ca80357d911f02d7038393f5659d43523
--> [internal] naming to 637423495437.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v3
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/c042z1wx9hs8t0h4i7zjvd

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

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13 (5.032s)
aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-stdin 637423495437.dkr.ecr.us-west-2.amazonaws.com

Login Succeeded

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13 (58.262s)
docker push 637423495437.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v3

The push refers to repository [637423495437.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app]
16bf2f260b78: Pushed
0b2c53ae7562: Layer already exists
3482d4cd60de: Layer already exists
370c0e783eaa: Layer already exists
a74bee0a4845: Layer already exists
c8f253ae7560: Layer already exists
a83d0a80a000: Layer already exists
v3: digest: sha256:76d0eb77e88e75e188bb94d0fa2df3772237b3afa91716d9db48ddf69f264 size: 1999

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (2.472s)
kubectl apply -f minikube.yaml

persistentvolumeclaim/mongo_persistent-storage unchanged
service/mongo-service unchanged
deployment.apps/mongo unchanged
deployment.apps/todo-app configured
service/todo-app-service unchanged

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13

```

- Verify the status of the rolling update process using **kubectl rollout status deployment/todo-app**.

```

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (1.302s)
kubectl rollout status deployment/todo-app

deployment "todo-app" successfully rolled out

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (1.218s)
kubectl get svc todo-app-service

NAME          TYPE        CLUSTER-IP      EXTERNAL-IP
todo-app-service   LoadBalancer  10.100.169.180   a481d2ce94e6e4a328e653cf618d8900-429554333.us-west-2.elb.amazonaws.com   PORT(S)        AGE
                                                               80:30282/TCP   74m

```

- Test the application

← → ⚡ Not Secure a481d2ce94e6e4a328e653cf618d8900-429554333.us-west-2.elb.amazonaws.com

Coding Studio randr97/randr97.gi... Meet - muv-svmp... Albert Job search toolkit... Top Amazon Ques... Matching Fuzzy Te...

All Bookmarks

ToDo Reminder

ALL Uncompleted Completed

Search Reference: Unique ID ▾ Search Task Search

To-Do LIST :

Status	Task Name	Description Name	Date	Priority	Remove	Modify
X	Reet 1	Reet 1	2024-03-18	Low !		

Add a Task

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

Create

Part 7: Health Monitoring

- Configure Kubernetes to handle liveness and readiness probes for the application.
- Monitor the health status of pods within the cluster.

```
base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (1.151s)
kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
flask-app-rc-l9k6b   1/1     Running   0          33m
flask-app-rc-snr7d   1/1     Running   0          33m
flask-app-rc-tnrz9   1/1     Running   0          33m
mongo-56dbc9457-2hpcs   1/1     Running   0          42m
todo-app-55d497dc55-t2vcr   1/1     Running   0          36s
todo-app-55d497dc55-v6tn2   1/1     Running   0          46s
```

- Validate the health monitoring system by deliberately introducing failures.
 - I deliberately changed the endpoint as **list1** to cause error

```
template:
  metadata:
    labels:
      app: todo-app
  spec:
    containers:
      - name: todo-app-container
        image: 637423495437.dkr.ecr.us-west-2.amazonaws.com/todo-flask-
        ports:
          - containerPort: 5000
        livenessProbe:
          httpGet:
            path: /list1
            port: 5000
          initialDelaySeconds: 10
          periodSeconds: 5
        readinessProbe:
          httpGet:
            path: /list1
            port: 5000
          initialDelaySeconds: 5
          periodSeconds: 5
```

```
base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (1.124s)
kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
flask-app-rc-l9k6b   1/1    Running   0          36m
flask-app-rc-snr7d   1/1    Running   0          36m
flask-app-rc-tnrz9   1/1    Running   0          36m
mongo-56dbcd9457-2hpcs   1/1    Running   0          45m
todo-app-55d497dc55-t2vcr   1/1    Running   0          3m31s
todo-app-55d497dc55-v6tn2   1/1    Running   0          3m41s
todo-app-76b99fffbf-gfzjc   0/1    Running   1 (1s ago)  22s
```

```
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  40s            default-scheduler  Successfully assigned default/todo-app-76b99fffbf-gfzjc to ip-192-168-33-199.us-west-2.compute.internal
  Normal   Pulled    28s            kubelet            Container Image "637423495437.dkr.ecr.us-west-2.amazonaws.com/todo-flask-app:v3" already present on machine
  Normal   Created   28s            kubelet            Created container todo-app-container
  Normal   Killing   28s            kubelet            Container todo-app-container failed liveness probe, will be restarted
  Normal   Started   19s            kubelet            Started container todo-app-container
  Warning  Unhealthy  5s (x8 over 35s)  kubelet            Readiness probe failed: HTTP probe failed with statuscode: 404
  Warning  Unhealthy  5s (x5 over 30s)  kubelet            Liveness probe failed: HTTP probe failed with statuscode: 404
```

- Revert back endpoints to list

```
base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (1.912s)
kubectl apply -f mntkube.yaml
persistentvolumeclaim/mongo-persistent-storage unchanged
service/mongo-service unchanged
deployment.apps/mongo unchanged
deployment.apps/todo-app configured
service/todo-app-service unchanged

base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13 * arn:aws:eks:us-west-2:637423495437:cluster/todo-cluster (1.44s)
kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
flask-app-rc-l9k6b   1/1    Running   0          38m
flask-app-rc-snr7d   1/1    Running   0          38m
flask-app-rc-tnrz9   1/1    Running   0          38m
mongo-56dbcd9457-2hpcs   1/1    Running   0          46m
todo-app-55d497dc55-t2vcr   1/1    Running   0          5m22s
todo-app-55d497dc55-v6tn2   1/1    Running   0          5m32s
```

Part 8:

- Add the Prometheus Community Helm repo

```
helm repo add prometheus-community
```

```
https://prometheus-community.github.io/helm-charts
```

```
helm repo update
```

```
helm upgrade prometheus prometheus-community/prometheus --namespace  
prometheus --set
```

```
alertmanager.persistentVolume.storageClass="standard",server.persistentVolume  
storageClass="standard" --reuse-values
```

- Check PVC POD

```
base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13 * minikube (0.104s)  
kubectl get pvc -n prometheus
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
prometheus-server	Bound	pvc-8a159492-4b33-42fc-a41c-8afbdd55af27	8Gi	RWO	standard	12s
storage-prometheus-alertmanager-0	Bound	pvc-9c96041b-cbfe-46fd-8b2a-8e4d19520bf4	2Gi	RWO	standard	6m31s

```
base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±13 * minikube (0.099s)  
kubectl get pods -n prometheus
```

NAME	READY	STATUS	RESTARTS	AGE
prometheus-alertmanager-0	1/1	Running	0	9m58s
prometheus-kube-state-metrics-75cd95986-m6rz9	1/1	Running	0	9m58s
prometheus-prometheus-node-exporter-zcnq6	1/1	Running	0	9m58s
prometheus-prometheus-pushgateway-76976dc66-mvgzj	1/1	Running	0	9m58s
prometheus-server-54886dcc45-xj6ft	2/2	Running	0	9m58s

- Write alertmanager-config.yaml and test-alert-rule.yaml files and run:

```
kubectl get statefulsets -n prometheus
```

```
base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±14 * minikube (0.212s)  
kubectl apply -f alertmanager-config.yaml
```

```
Warning: resource configmaps/prometheus-alertmanager is missing the kubectl.kubernetes.io/last-applied-con  
declaratively by either kubectl create --save-config or kubectl apply. The missing annotation will be pat  
configmap/prometheus-alertmanager configured
```

```
base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±14 * minikube (0.096s)  
kubectl rollout restart statefulset alertmanager -n prometheus
```

```
Error from server (NotFound): statefulsets.apps "alertmanager" not found
```

```
base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±14 * minikube (0.093s)  
kubectl get statefulsets -n prometheus
```

NAME	READY	AGE
prometheus-alertmanager	1/1	46m

```
base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±14 * minikube (0.105s)  
kubectl rollout restart statefulset prometheus-alertmanager -n prometheus
```

```
statefulset.apps/prometheus-alertmanager restarted
```

```
kubectl apply -f test-alert-rule.yaml
```

```
base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±15 * minikube (0.363s)
kubectl apply -f test-alert-rule.yaml

prometheusrule.monitoring.coreos.com/test-alert created
```

- Check Dashboard

Targets

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
https://192.168.49.2:8443/metrics	UP	instance="192.168.49.2:8443" job="kubernetes-apiservers"	41.967s ago	93.227ms	

kubernetes-nodes (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
https://kubernetes.default.svc/api/v1/nodes/minikube/proxy/metrics	UP	beta_kubernetes_io_arch="amd64" beta_kubernetes_io_os="linux" instances="minikube" job=kubernetes-nodes kubernetes_io_arch="amd64" kubernetes_io_hostname="minikube" kubernetes_io_os="linux" minikube_k8s_io_commit="8220a6eb10f04d75f27c27154cef975f050512d" minikube_k8s_io_name="minikube" minikube_k8s_io_primary=true minikube_k8s_io_updated_at="2024_04_18T10_56_58_070" minikube_k8s_io_version="v1.32.0"	5.445s ago	985.755ms	

kubernetes-nodes-cadvisor (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
https://kubernetes.default.svc/api/v1/nodes/minikube/proxy/metrics/cadvisor	UP	beta_kubernetes_io_arch="amd64" beta_kubernetes_io_os="linux" instances="minikube" job=kubernetes-nodes-cadvisor kubernetes_io_arch="amd64" kubernetes_io_hostname="minikube" kubernetes_io_os="linux" minikube_k8s_io_commit="8220a6eb10f04d75f27c27154cef975f050512d" minikube_k8s_io_name="minikube" minikube_k8s_io_primary=true minikube_k8s_io_updated_at="2024_03_18T10_56_58_070" minikube_k8s_io_version="v1.32.0"	36.353s ago	48.238ms	

kubernetes-service-endpoints (4/4 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.49.2:9100/metrics	UP	app_kubernetes_io_components="metrics" app_kubernetes_io_instances="prometheus" app_kubernetes_io_managed_by="Helm" app_kubernetes_io_name="prometheus-node-exporter" app_kubernetes_io_port_of="prometheus-node-exporter" app_kubernetes_io_version="1.7.0" helm_sh_chart="prometheus-node-exporter-4.30.3" instances="192.168.49.2:9100" https_kubernetes_service_endpoints="true"	56.588s ago	202.347ms	

Rules

Rule	Interval: 1m 0s	31.768s ago	0.221ms
kubernetes-replication			
alert: DeploymentReplicaFailure expr: kube_deployment_status_replicas_available(namespace="default") < kube_deployment_spec_replicas(namespace="default") labels: severity: critical annotations: description: Deployment {{ \$labels.deployment }} has less available replicas than desired. summary: Deployment Replica Failure in {{ \$labels.namespace }} namespace	OK	31.769s ago	0.208ms

- Now to create a scenario to test failure, we can delete a pod
kubectl delete pod todo-app-56fc88ddd6-sbxhc -n default

```
base ~/Desktop/Studies/Spring 2024/Cloud Computing/Assignment/2 git:(main)±15 * minikube (0.611s)
kubectl delete pod todo-app-56fc88ddd6-sbxhc -n default
pod "todo-app-56fc88ddd6-sbxhc" deleted
```

