

Assignment 2- Steps for implementation

Part 3: Deploying app on minikube

- Install minikube
- Start a minikube cluster

```
> minikube status
minikube
type: Control Plane
host: Stopped
kubelet: Stopped
apiserver: Stopped
kubeconfig: Stopped

> minikube start
🌟 minikube v1.34.0 on Darwin 15.1
⭐️ Using the docker driver based on existing profile
👍 Starting "minikube" primary control-plane node in "minikube" cluster
🌐 Pulling base image v0.0.45 ...
🔄 Restarting existing docker container for "minikube" ...
🛠️ Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
🌐 Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟 Enabled addons: default-storageclass, storage-provisioner
🎉 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
~ >
```

- Run the following commands to deploy on minikube:
kubectl apply -f kube/mongo-config.yml
kubectl apply -f kube/mongo.yml
kubectl apply -f kube/webapp.yml

```
> kubectl get all
NAME                                         READY   STATUS    RESTARTS   AGE
pod/mongo-deployment-77cd44d469-75ft5      1/1     Running   1 (4d18h ago)   5d1h
pod/webapp-deployment-9cbdbd55b-ddj4w       1/1     Running   0           6s

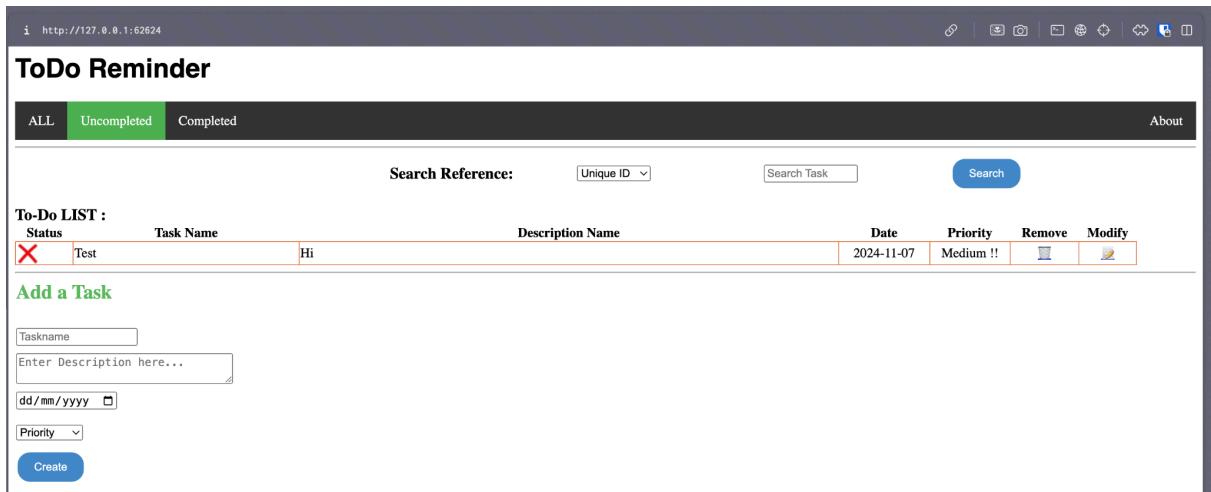
NAME          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP   5d21h
service/mongo-service   ClusterIP   10.103.169.148 <none>        27017/TCP  5d21h
service/webapp-service   LoadBalancer 10.107.204.38  <pending>     80:30100/TCP 5d21h

NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/mongo-deployment   1/1     1           1           5d21h
deployment.apps/webapp-deployment  1/1     1           1           5d21h

NAME          DESIRED   CURRENT   READY   AGE
replicaset.apps/mongo-deployment-77cd44d469  1         1         1         5d1h
replicaset.apps/webapp-deployment-9cbdbd55b   1         1         1         5d20h
```

- Run minikube service webapp-service –url to get the url

```
> minikube service webapp-service --url
http://127.0.0.1:62624
! Because you are using a Docker driver on darwin, the terminal needs to be open to run it.
```



Part 4: Deploying the Application on AWS EKS

- Create AWS EKS cluster eks-app with t3.medium as the instance type and 2 worker nodes. Also add the EFS CSI Driver add on on the cluster.
- Run the command to connect to the eks cluster : aws eks update-kubeconfig --region us-east-2 --name eks-app.
- Create an EFS file storage for mounting PV. ANd copy the efs id from it to the sc.yaml
- Run the commands to deploy the Kubernetes app on the eks cluster :


```
kubectl apply -f sc.yaml
kubectl apply -f flask-deployment.yaml
kubectl apply -f mongo-deployment.yaml
kubectl apply -f flask-service.yaml
kubectl apply -f mongo-service.yaml
```
- Verify that the pods and services are functional

```
rachit@Rachits-MacBook-Pro flask-todo-app-k8s % kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
flask-deployment-8649f4666-t6mjn   1/1    Running   0          95s
mongo-deployment-fbbcc68cb-h66bv   1/1    Running   0          26s
rachit@Rachits-MacBook-Pro flask-todo-app-k8s % kubectl get service
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP
        PORT(S)      AGE
flask-service   LoadBalancer   10.100.194.20   a646420dc36114d91a989828debe7067-
1384764920.us-east-2.elb.amazonaws.com   80:32428/TCP   32s
kubernetes     ClusterIP     10.100.0.1      <none>
                                         443/TCP      11m
mongo         ClusterIP     10.100.135.0    <none>
                                         27017/TCP     24s
```

- Ensure deployments are functional:

```
rachit@Rachits-MacBook-Pro flask-todo-app-k8s % kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
flask-deployment   1/1       1           1          11m
mongo-deployment   1/1       1           1          10m
```

- Run the command: kubectl get service, to retrieve external ip on which Kubernetes app is deployed.

```
rachit@Rachits-MacBook-Pro flask-todo-app-k8s % kubectl get service
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP
        PORT(S)      AGE
flask-service   LoadBalancer   10.100.194.20   a646420dc36114d91a989828debe7067-
1384764920.us-east-2.elb.amazonaws.com   80:32428/TCP   32s
kubernetes     ClusterIP     10.100.0.1      <none>
                                         443/TCP      11m
mongo         ClusterIP     10.100.135.0    <none>
                                         27017/TCP     24s
```

- Test the application:

The screenshot shows a web browser window with the following details:

- Header:** Includes links for 'Not Secure', 'a646420dc36114d91a989828debe7067-1384764920.us-east-2.elb.amazonaws.com', 'The Company Me...', 'Warren Buffett - H...', 'VITCC-INTRANET', 'Thank you for do...', 'Rocky.1-6.The.Co...', 'Ice.Princess.2005...', 'iCloud Contacts', 'FFCS - Registration', 'Subtitles', 'VPN', 'Update', and 'All Bookmarks'.
- Title:** ToDo Reminder
- Navigation:** A dark navigation bar with tabs: ALL (selected), Uncompleted (highlighted in green), and Completed.
- Message:** No Tasks in the List !!
- Form:**
 - Taskname:
 - Enter Description here...:
 - Date: dd/mm/yyyy
 - Priority:
 - Create:

Part 5: Replication controller feature

- Create a file named flask-replication-controller.yaml with 2 replicas.
- Run the command: kubectl apply -f flask-replication-controller.yaml to create the replication controller.
- Run kubectl get replicationcontroller:

```
rachit@Rachits-MacBook-Pro flask-todo-app-k8s % kubectl get replicationcontroller
NAME          DESIRED  CURRENT  READY  AGE
flask-replication-controller  2        2        2      12s
rachit@Rachits-MacBook-Pro flask-todo-app-k8s % kubectl get pods -l app=flask-app
```

- Run kubectl get pods -l app=flask-app to see 2 replicas of the controller running:

```
rachit@Rachits-MacBook-Pro flask-todo-app-k8s % kubectl get pods -l app=flask-app
NAME          READY  STATUS  RESTARTS  AGE
flask-deployment-795df447cd-mrn9b  1/1   Running   0      3h36m
flask-replication-controller-d5n6w  1/1   Running   0      32s
flask-replication-controller-lqnrc  1/1   Running   0      32s
```

- Deleting one of the replicas to check if another one is automatically created by the controller:

```
[rachit@Rachits-MacBook-Pro flask-todo-app-k8s % kubectl delete pod flask-replication-controller-d5n6w
pod "flask-replication-controller-d5n6w" deleted
^C
rachit@Rachits-MacBook-Pro flask-todo-app-k8s % kubectl get pods -l app=flask-app
NAME          READY  STATUS  RESTARTS  AGE
flask-deployment-795df447cd-mrn9b  1/1   Running   0      3h38m
flask-replication-controller-2t2zz  1/1   Running   0      35s
flask-replication-controller-lqnrc  1/1   Running   0      108s
```

- Scaling the app by increasing the number of replicas to 3

```
rachit@Rachits-MacBook-Pro flask-todo-app-k8s % kubectl apply -f flask-replication-controller.yaml
replicationcontroller/flask-replication-controller configured
rachit@Rachits-MacBook-Pro flask-todo-app-k8s % kubectl get pods -l app=flask-app
NAME                               READY   STATUS    RESTARTS   AGE
flask-deployment-795df447cd-mrn9b  1/1    Running   0          3h39m
flask-replication-controller-2t2zz  1/1    Running   0          85s
flask-replication-controller-lqnrc  1/1    Running   0          2m38s
flask-replication-controller-pw79v  1/1    Running   0          8s
```

The replica flask-replication-controller-pw79v has been recently created.

Part6: Rolling update strategy:

- Inside flask-deployment.yaml add these three lines of code under spec:
`rollingUpdate:`
 `maxUnavailable: 1`
 `maxSurge: 1`

This ensures that the maximum number of unavailable pods during the update is 1 and only 1 extra pod can be created during the update.

Save the file as flask-update-deployment

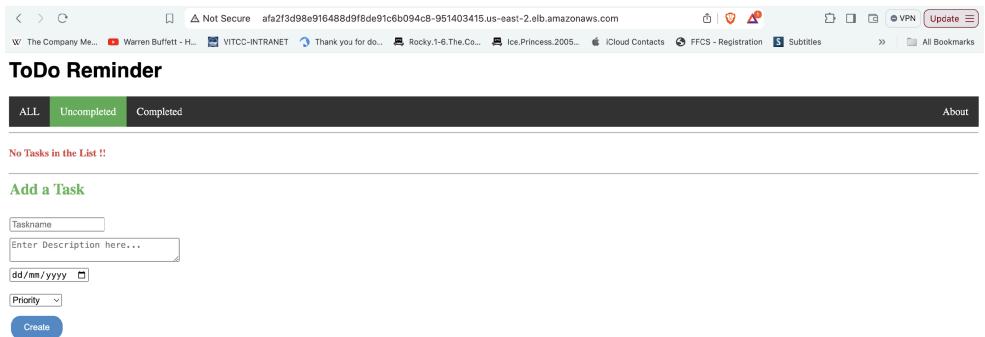
- Update the image version to v2 from v1. Run the command :
`kubectl set image deployment/flask-deployment flask-container=rachit798/flask-todo-app:v2`
- Run kubectl apply -f flask-update-deployment.yaml
- To check status of rolling update progress run the command :
`kubectl rollout status deployment flask-deployment`

```
rachit@Rachits-MacBook-Pro flask-todo-app-k8s % kubectl rollout status deployment flask-deployment
deployment "flask-deployment" successfully rolled out
```

- Verifying that pods are running after update :

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATE
flask-deployment-85448bdfdc-66rrz	1/1	Running	0	22m	172.31.32.147	ip-172-31-38-87.us-east-2.compute.internal	<none>	<none>
flask-deployment-85448bdfdc-k1t5z	1/1	Running	0	22m	172.31.6.104	ip-172-31-13-146.us-east-2.compute.internal	<none>	<none>
flask-deployment-85448bdfdc-qdv96	1/1	Running	0	22m	172.31.35.198	ip-172-31-38-87.us-east-2.compute.internal	<none>	<none>

- Testing application:



Part 7: Health monitoring:

- Configure probe endpoints as /healthz and /ready for the health and readiness probes and add the probe parameters under spec in flask-deployment.yaml.
- Monitoring health of pods and checking for success/failure of probe:

```
rachit@Rachits-MacBook-Pro ~ % kubectl describe pod flask-deployment-5fcfd5768d7-9v681
Name:           flask-deployment-5fcfd5768d7-9v681
Namespace:      default
Priority:       0
Service Account: default
Node:          ip-172-31-4-14.us-east-2.compute.internal/172.31.4.14
Start Time:    Tue, 05 Nov 2024 19:08:04 -0500
Labels:         app=flask-app
                pod-template-hash=5fcfd5768d7
Annotations:   <none>
Status:        Running
IP:            172.31.13.45
IPs:
  IP:          172.31.13.45
Controlled By: ReplicaSet/flask-deployment-5fcfd5768d7
Containers:
  flask-container:
    Container ID:  containerd://5353e17e9hd5105a4ded0c748870015f341ba2b7c8ccfd68aa62c73d190e9081
    Image:          docker.io/rachit798/flask-todo-app:v1
    Image ID:      docker.io/rachit798/flask-todo-app@sha256:18c82b0a96e1c31ebc9b314713d2d34bfe850b33cf2e92bc55816068ef9f4f55
    Port:          5000/TCP
    Host Port:    0/TCP
    State:        Running
    Started:     Tue, 05 Nov 2024 19:08:05 -0500
    Ready:        True
    Restart Count: 0
    Liveness:     http-get http://:5000/healthz delay=15s timeout=3s period=5s #success=1 #failure=3
    Readiness:    http-get http://:5000/ready delay=15s timeout=3s period=5s #success=1 #failure=3
    Environment:
      MONGO_HOST: mongo
      MONGO_PORT: 27017
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-qnztr (ro)
Conditions:
  Type        Status
  PodReadyToStartContainers  True
  Initialized  True
  Ready        True
  ContainersReady  True
  PodScheduled  True
Volumes:
  kube-api-access-qnztr:
    Type:       Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:        kube-root-ca.crt
    ConfigMapOptional:    <nil>
    DownwardAPI:          true
Events:
  Type      Reason     Age   From           Message
  ----      ----     --   --            --
  Normal    Scheduled  23s  default-scheduler  Successfully assigned default/flask-deployment-dc795dc5-8w4tj to ip-172-31-11-236.us-east-2.compute.internal
  Normal    Pulling    23s  kubelet        Pulling image "docker.io/rachit798/flask-todo-app:v1"
  Normal    Pulled    23s  kubelet        Successfully pulled image "docker.io/rachit798/flask-todo-app:v1" in 176ms (176ms including waiting). Image size: 43188
  bytes.
  Normal    Created    23s  kubelet        Created container flask-container
  Normal    Started    23s  kubelet        Started container flask-container
  Warning  Unhealthy  3s   kubelet        Liveness probe failed: HTTP probe failed with statuscode: 500
  Warning  Unhealthy  3s   kubelet        Readiness probe failed: HTTP probe failed with statuscode: 503
```

Liveness Probe: Ensures that the pod is alive and healthy. If it fails repeatedly, Kubernetes will attempt to restart the pod.

Readiness Probe: Ensures that the pod is ready to handle traffic. If it fails, Kubernetes will stop routing traffic to the pod until the readiness probe passes.

- Testing health monitoring system by causing a failure:

```
Events:
  Type      Reason     Age   From           Message
  ----      ----     --   --            --
  Normal    Scheduled  23s  default-scheduler  Successfully assigned default/flask-deployment-dc795dc5-8w4tj to ip-172-31-11-236.us-east-2.compute.internal
  Normal    Pulling    23s  kubelet        Pulling image "docker.io/rachit798/flask-todo-app:v1"
  Normal    Pulled    23s  kubelet        Successfully pulled image "docker.io/rachit798/flask-todo-app:v1" in 176ms (176ms including waiting). Image size: 43188
  bytes.
  Normal    Created    23s  kubelet        Created container flask-container
  Normal    Started    23s  kubelet        Started container flask-container
  Warning  Unhealthy  3s   kubelet        Liveness probe failed: HTTP probe failed with statuscode: 500
  Warning  Unhealthy  3s   kubelet        Readiness probe failed: HTTP probe failed with statuscode: 503
```

Part 8: Alerting

We first install helm and add the prometheus repository to it. After this we can install the prometheus operator using a helm chart deployment.

```
helm repo add prometheus-community
https://prometheus-community.github.io/helm-charts
helm repo update
helm install assign2
prometheus-community/kube-prometheus-stack
```

```

> kubectl get all
NAME                                         READY   STATUS    RESTARTS   AGE
pod/alertmanager-assign2-kube-prometheus-st-alertmanager-0   2/2     Running   0          164m
pod/assign2-grafana-67b6896dc4-llnvj           3/3     Running   0          164m
pod/assign2-kube-prometheus-st-operator-7d67c885f-wtdx4   1/1     Running   0          164m
pod/assign2-kube-state-metrics-755fc5fff-kh6lk   1/1     Running   0          164m
pod/assign2-prometheus-node-exporter-2w7v       1/1     Running   0          164m
pod/assign2-prometheus-node-exporter-z7rps      1/1     Running   0          164m
pod/flask-deployment-867d6c64d-9y8z            1/1     Running   0          2m48s
pod/flask-deployment-867d6c64d-ffhv            1/1     Running   0          2m48s
pod/mongo-deployment-1bbcc68cb-844ps          1/1     Running   0          46h
pod/prometheus-assign2-kube-prometheus-st-prometheus-0  2/2     Running   0          164m

NAME                                         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
service/alertmanager-operated   ClusterIP  None         <none>        9093/TCP, 9094/TCP, 9094/UDP   164m
service/assign2-grafana          ClusterIP  10.100.190.236  <none>        80/TCP           164m
service/assign2-kube-prometheus-st-alertmanager   ClusterIP  10.100.222.198  <none>        9093/TCP, 8080/TCP   164m
service/assign2-kube-prometheus-st-operator   ClusterIP  10.100.199.51  <none>        443/TCP          164m
service/assign2-kube-prometheus-st-prometheus   ClusterIP  10.100.157.169 <none>        9090/TCP, 8080/TCP   164m
service/assign2-kube-state-metrics          ClusterIP  10.100.52.181  <none>        8080/TCP          164m
service/assign2-prometheus-node-exporter      ClusterIP  10.100.147.193 <none>        9100/TCP          164m
service/flask-service                   LoadBalancer 10.100.121.120  a880ba66b22ae44939cb56ed3865d12c-75573875.us-east-1.elb.amazonaws.com  80:32646/TCP   3d18h
service/kubernetes                     ClusterIP  10.100.0.1     <none>        443/TCP          3d20h
service/mongo                          ClusterIP  10.100.74.61   <none>        27017/TCP         3d18h
service/prometheus-operated           ClusterIP  None         <none>        9090/TCP          164m

NAME             DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/assign2-prometheus-node-exporter  2          2         2        2           2           kubernetes.io/os=linux   164m

NAME             READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/assign2-grafana      1/1     1           1           164m
deployment.apps/assign2-kube-prometheus-st-operator 1/1     1           1           164m
deployment.apps/assign2-kube-state-metrics      1/1     1           1           164m
deployment.apps/flask-deployment     3/3     3           3           3d18h
deployment.apps/mongo-deployment     1/1     1           1           3d18h

NAME             DESIRED   CURRENT   READY   AGE
replicaset.apps/assign2-grafana-67b6896dc4  1          1         1        164m
replicaset.apps/assign2-kube-prometheus-st-operator-7d67c885f  1          1         1        164m
replicaset.apps/assign2-kube-state-metrics-755fc5fff  1          1         1        164m
replicaset.apps/flask-deployment-867d6c64d  3          3         3        5h43m
replicaset.apps/mongo-deployment-fbcc68cb      1          1         1        46h

NAME             READY   AGE
statefulset.apps/alertmanager-assign2-kube-prometheus-st-alertmanager  1/1     164m
statefulset.apps/prometheus-assign2-kube-prometheus-st-prometheus        1/1     164m

```

Once we have prometheus up and running, we have to set up a prometheus rule that would trigger an alert. The config for this alert is under `sampleRule.yaml`. This creates an external resource named `flaskapp-rules` that is picked up by the `kube-prometheus-stack` operator.

`kubectl apply -f sampleRule.yaml`

```

> kubectl get PrometheusRule
NAME          AGE
flaskapp-rules 179m
~ > 

```

flaskapp.rule	Interval: 30.0s	1.767s ago	0.858ms	
Rule <pre> alert: FlaskAppDown expr: max_over_time(kube_pod_container_status_waiting_reason{job="kube-state-metrics",namespace="default",pod=~"flask-deployment-*",reason="CrashLoopBackOff"}[5m]) >= 1 for: 5s labels: severity: critical annotations: description: Pod {{ \$labels.namespace }}/{{ \$labels.pod }} ({{ \$labels.container }}) has been in CrashLoopBackOff state for more than 5 seconds. summary: FlaskApp Pod is crash looping </pre>	State: OK	Error:	Last Evaluation: 1.768s ago	Evaluation Time: 0.833ms

When the conditions in the `expr` statement are met, an alert is fired. This alert is then picked up by the AlertManager. At this stage we need to configure the AlertManager to send a slack notification when it receives the alert. This configuration is

attached under alert-manager.yaml and is registered with the provider using kubectl external resource.

```
kubectl apply -f secret.yml  
kubectl apply -f alert-manager.yaml
```

The screenshot shows the Alertmanager interface with three alerts listed. Each alert entry includes a timestamp, an 'Info' button, a 'Source' link, a 'Silence' button, and a 'Link' button. Below each entry is a detailed alert configuration card. The first alert is for 'FlaskAppDown' with pod 'flask-deployment-86c7d6c64d-bvhfw', prometheus 'default/assign2-kube-prometheus-st-prometheus', reason 'CrashLoopBackOff', service 'assign2-kube-state-metrics', severity 'critical', and uid '83bba25c-c495-45d3-ad6f-4316b226e037'. The second and third alerts are similar, with slight differences in their configuration details.

```
❯ kubectl get AlertmanagerConfig  
NAME      AGE  
config-slack  34m  
  
Alertmanager  Alerts  Silences  Status  Settings  Help  New Silence  
  
Filter  Group  Receiver: All  Silenced  Inhibited  
Custom matcher, e.g. env="production" +  ✎ Silence  
  
+ Expand all groups  
- default/config-slack/webhook  job="kube-state-metrics" +  3 alerts  
  
2024-11-10T20:01:43.197Z  + Info  ↗ Source  ✎ Silence  % Link  
alertname="FlaskAppDown"  container="flask-container"  endpoint="http"  instance="172.31.13.210:8080"  namespace="default"  pod="flask-deployment-86c7d6c64d-bvhfw"  prometheus="default/assign2-kube-prometheus-st-prometheus"  reason="CrashLoopBackOff"  service="assign2-kube-state-metrics"  severity="critical"  uid="83bba25c-c495-45d3-ad6f-4316b226e037"  
  
2024-11-10T20:01:43.197Z  + Info  ↗ Source  ✎ Silence  % Link  
alertname="FlaskAppDown"  container="flask-container"  endpoint="http"  instance="172.31.13.210:8080"  namespace="default"  pod="flask-deployment-86c7d6c64d-ss9c4"  prometheus="default/assign2-kube-prometheus-st-prometheus"  reason="CrashLoopBackOff"  service="assign2-kube-state-metrics"  severity="critical"  uid="6c874b43-ecb3-4651-b3b7-e1272ffcc6f"  
  
2024-11-10T20:01:43.197Z  + Info  ↗ Source  ✎ Silence  % Link  
alertname="FlaskAppDown"  container="flask-container"  endpoint="http"  instance="172.31.13.210:8080"  namespace="default"  pod="flask-deployment-86c7d6c64d-zphgl"  prometheus="default/assign2-kube-prometheus-st-prometheus"  reason="CrashLoopBackOff"  service="assign2-kube-state-metrics"  severity="critical"  uid="6979e00a-5c91-4448-b742-fa6f26dbc540"
```

In order to create this error we have pushed the following faulty code:

```
1 app.py @@ health_check
2     heading = "Todo Reminder"
3     def is_db_connected():
4         try:
5             # Try a simple command to check if the database is reachable
6             client.admin.command('ping')
7             return True
8         except Exception as e:
9             return False
10    # Add liveness probe endpoint
11    @app.route("/healthz")
12    def health_check():
13        return "Healthy", 200 # Simple check
14    +
15    @app.route("/failure")
16        You, 1 second ago + Uncommitted changes
17    def health_check():
18        return "Failed", 500 # Simple check
19
20    # Add readiness probe endpoint
21    @app.route("/ready")
22    def readiness_check():
23        if is_db_connected():
24            return "Ready", 200
25        else:
26            return "Not Ready", 503
27
28
29 TERMINAL PORTS GITLENS PROBLEMS ⑥ OUTPUT DEBUG CONSOLE COMMENTS
30
31 => [web internal] load build definition from Dockerfile
32 =>> transferring dockerfile: 363B
33 => [web internal] load metadata for docker.io/library/python:3.9-slim
34 => [web auth] library/python:pull took for registry-1.docker.io
35 => [web internal] load build context
36 => transferring context: 1.164B
37 => [web internal] FROM docker.io/library/python:3.9-slim@sha256:7a9cd42786c174cdcf57888bab9ae3b6551323a7ddbc2a89ad65b20a28fbfbe
38 => [web internal] load build context
39 => writing manifest file for image: 4.41kB
40 => CACHED [web 3/5] WORKDIR /app
41 => CACHED [web 3/5] COPY requirements.txt requirements.txt
42 => CACHED [web 3/5] RUN pip install --no-cache-dir -r requirements.txt
43 => [web 3/5] COPY . /app
44 => [web] exporting to image
45 => => exporting layers
46 => => writing image sha256:32a1fa40f7f3bc8a24d95f9ee53f922e07c1bdd42a0b@ea16559152b4bc748
47 => => writing manifest file for image: 1.01kB
48 => => writing config file for image: 1.01kB
49 => => writing layer sha256:78867e: Pushed
50 => docker tag cloud-assignment2-web singhdivyansh/cloud-assignment2-web
51 => docker image push singhdivyansh/cloud-assignment2-web
52 Using default tag: latest
53 The push refers to repository [docker.io/singhdivyansh/cloud-assignment2-web]
54 10a0fd78867e: Pushed
55 12e82d062860: Layer already exists
56 12e82d062860: Layer already exists
57 9142c272ef1: Layer already exists
58 086feaf88e98: Layer already exists
59 19f5acfc4f88: Layer already exists
60 19f5acfc4f88: Layer already exists
61 98853c909d3: Layer already exists
62 latest: digest: sha256:10415e243d34ea8a0b125e498afdf90256d75cd13aed257bbad7c4ab44fe5d97377 size: 1994
63
64 /Programming/cloud-comg/Cloud-Assignment2/flask-todo-app-k8s main+1|14 73 >
```

```

> kubectl get all
NAME                                         READY   STATUS    RESTARTS   AGE
pod/alertmanager-assign2-kube-prometheus-st-alertmanager-0   2/2     Running   0          3h49m
pod/assign2-grafana-67b6896dc4-llnvj                3/3     Running   0          3h49m
pod/assign2-kube-prometheus-st-operator-7d67c885f-wtdx4   1/1     Running   0          3h49m
pod/assign2-kube-state-metrics-755fc5fff-kh6lk        1/1     Running   0          3h49m
pod/assign2-prometheus-node-exporter-2v7zv           1/1     Running   0          3h49m
pod/assign2-prometheus-node-exporter-z7rps          1/1     Running   0          3h49m
pod/flask-deployment-86c7d6c64d-dvhfw             1/1     Running   9 (42m ago) 58m
pod/flask-deployment-86c7d6c64d-ss9c4             1/1     Running   9 (42m ago) 58m
pod/flask-deployment-86c7d6c64d-zpngl            1/1     Running   9 (42m ago) 58m
pod/mongo-deployment-fbbc68cb-844ps            1/1     Running   0          47h
pod/prometheus-assign2-kube-prometheus-st-prometheus-0  2/2     Running   0          3h49m

NAME                                         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
service/alertmanager-operated   ClusterIP  None         <none>        9093/TCP,9094/TCP, 9094/UDP  3h49m
service/assign2-grafana        ClusterIP  10.100.190.236 <none>        80/TCP           3h49m
service/assign2-kube-prometheus-st-alertmanager   ClusterIP  10.100.222.198 <none>        9093/TCP,8080/TCP  3h49m
service/assign2-kube-prometheus-st-operator      ClusterIP  10.100.199.51  <none>        443/TCP          3h49m
service/assign2-kube-prometheus-st-prometheus   ClusterIP  10.100.157.169 <none>        9090/TCP,8080/TCP  3h49m
service/assign2-kube-state-metrics            ClusterIP  10.100.52.181 <none>        8880/TCP          3h49m
service/assign2-prometheus-node-exporter       ClusterIP  10.100.147.193 <none>        9100/TCP          3h49m
service/flask-service                      LoadBalancer 10.100.121.120 a880ba6b22ae44939cb56ed3865d12c-75573875.us-east-1.elb.amazonaws.com  80:32646/TCP  3d19h
service/kubernetes                       ClusterIP  10.100.0.1   <none>        443/TCP          3d21h
service/mongo                            ClusterIP  10.100.74.61  <none>        27017/TCP          3d19h
service/prometheus-operated              ClusterIP  None         <none>        9090/TCP          3h49m

NAME                                         DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/assign2-prometheus-node-exporter  2         2         2         2           2           kubernetes.io/os:linux  3h49m

NAME                                         READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/assign2-grafana               1/1     1           1           3h49m
deployment.apps/assign2-kube-prometheus-st-operator  1/1     1           1           3h49m
deployment.apps/assign2-kube-state-metrics      1/1     1           1           3h49m
deployment.apps/flask-deployment              3/3     3           3           3d19h
deployment.apps/mongo-deployment              1/1     1           1           3d19h

NAME                                         DESIRED   CURRENT   READY   AGE
replicaset.apps/assign2-grafana-67b6896dc4  1         1         1         3h49m
replicaset.apps/assign2-kube-prometheus-st-operator-7d67c885f  1         1         1         3h49m
replicaset.apps/assign2-kube-state-metrics-755fc5fff  1         1         1         3h49m
replicaset.apps/flask-deployment-86c7d6c64d  3         3         3         6h48m
replicaset.apps/mongo-deployment-fbbc68cb  1         1         1         47h

NAME                                         READY   AGE
statefulset.apps/alertmanager-assign2-kube-prometheus-st-alertmanager  1/1     3h49m
statefulset.apps/prometheus-assign2-kube-prometheus-st-prometheus  1/1     3h49m

```

Since the method `health_check` is registered with two separate routes, it will crash the app on startup, creating the alert we need to test our notification system.