

CLOUD COMPUTING

ASSIGNMENT - 2

EKS Cluster Nodes

EKS cluster worker nodes successfully provisioned and in Ready state. Two t3.medium EC2 instances running Kubernetes version 1.31 in the us-east-1 region, demonstrating successful cluster creation and node registration.

```
eks-development --zsh -- 127x43
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl get nodes
NAME           STATUS  ROLES   AGE    VERSION
ip-192-168-3-98.ec2.internal  Ready   <none>  50m   v1.31.13-eks-113cf36
ip-192-168-53-207.ec2.internal  Ready   <none>  50m   v1.31.13-eks-113cf36
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development %
```

All Kubernetes Resources

Complete view of deployed Kubernetes resources showing deployments, services, pods, and replica sets. The flask-todo-app deployment has 2/2 replicas available, MongoDB has 1/1 replica, and both services (flask-todo-service as NodePort, mongodb-service as ClusterIP) are successfully configured.

```
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl get all
NAME                           READY   STATUS    RESTARTS   AGE
pod/flask-todo-app-78dbc4c7bb-n8zc6  1/1    Running   0          30m
pod/flask-todo-app-78dbc4c7bb-rlqd9  1/1    Running   0          30m
pod/mongodb-8f797cf5b-6wsbv        1/1    Running   0          47m

NAME              TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
service/flask-todo-service  NodePort    10.100.96.44  <none>        80:30080/TCP  46m
service/kubernetes     ClusterIP   10.100.0.1    <none>        443/TCP       59m
service/mongodb-service ClusterIP   10.100.173.141 <none>        27017/TCP    47m

NAME            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/flask-todo-app  2/2     2           2           46m
deployment.apps/mongodb         1/1     1           1           47m

NAME            DESIRED   CURRENT   READY   AGE
replicaset.apps/flask-todo-app-78dbc4c7bb  2        2        2        46m
replicaset.apps/mongodb-8f797cf5b        1        1        1        47m
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development %
```

Application URL in Browser

Flask To-Do application successfully accessible via public URL using NodePort service on port 30080. The application homepage loads correctly, demonstrating successful external access configuration through AWS security groups and Kubernetes NodePort service.

ToDo Reminder

ALL Uncompleted Completed About

No Tasks in the List !!

Add a Task

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

Priority

Application with Multiple Tasks

Functional demonstration of the To-Do application with multiple tasks created. Shows task management features including task name, description, due date, priority levels, and filtering options (All/Uncompleted/Completed). Data persists in MongoDB, demonstrating successful database connectivity and CRUD operations.

ToDo Reminder

ALL Uncompleted Completed About

To-Do LIST :

Status	Task Name	Description Name	Date	Priority	Remove	Modify
X	Todo - 1	Testing...	2025-10-22	Low !	<input type="button" value=""/>	<input type="button" value=""/>
X	Cloud Assignment	EKS deployment	2025-10-26	Medium !!	<input type="button" value=""/>	<input type="button" value=""/>
X	Cloud Assignment	Verify deployment	2025-10-26	Medium !!	<input type="button" value=""/>	<input type="button" value=""/>

Add a Task

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

Priority

Task Marked as Complete

Task completion functionality demonstrated with a task marked as complete (indicated by checkmark/strikethrough). Shows the application's ability to update task status in MongoDB and reflect changes in the user interface in real-time, confirming successful database write operations.

ToDo Reminder

ALL Uncompleted Completed About

Search Reference: Unique ID Search

To-Do LIST :

Status	Task Name	Description Name	Date	Priority	Remove	Modify
✓	Todo - 1	Testing...	2025-10-22	Low !		
✗	Cloud Assignment	EKS deployment	2025-10-26	Medium !!		
✗	Cloud Assignment	Verify deployment	2025-10-26	Medium !!		

Add a Task

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

Create

Deployment Details

Detailed deployment configuration showing Flask application deployment specifications including container image (yogas007/flask-todo-app:latest), environment variables (MONGO_URI, FLASK_ENV), resource limits (CPU: 200m, Memory: 256Mi), replica count (2), and rolling update strategy (maxUnavailable: 1, maxSurge: 1).

```
eks-development --zsh -- 127x43
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl describe deployment flask-todo-app
Name:           flask-todo-app
Namespace:      default
CreationTimestamp: Fri, 31 Oct 2025 09:18:50 -0400
Labels:          app=flask-todo-app
Annotations:    deployment.kubernetes.io/revision: 1
Selector:        app=flask-todo-app
Replicas:       2 desired | 2 updated | 2 total | 2 available | 0 unavailable
StrategyType:   RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 1 max unavailable, 1 max surge
Pod Template:
  Labels:  app=flask-todo-app
           version=v1
  Containers:
    flask-todo-app:
      Image:      yogas007/flask-todo-app:latest
      Port:       5000/TCP
      Host Port:  0/TCP
      Limits:
        cpu:      200m
        memory:   256Mi
      Requests:
        cpu:      100m
        memory:   128Mi
      Environment:
        MONGO_URI:  mongodb://admin:password123@mongodb-service:27017/
        FLASK_ENV:   production
      Mounts:      <none>
      Volumes:     <none>
      Node-Selectors: <none>
      Tolerations:  <none>
  Conditions:
    Type     Status  Reason
    ----  -----
    Progressing  True    NewReplicaSetAvailable
    Available    True    MinimumReplicasAvailable
  OldReplicaSets: <none>
  NewReplicaSet:  flask-todo-app-78dbc4c7bb (2/2 replicas created)
  Events:        <none>
```

Pods auto-created after deletion

```
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl get pods
NAME READY STATUS RESTARTS AGE
flask-todo-app-78dbc4c7bb-mz6b8 1/1 Running 0 7s
flask-todo-app-78dbc4c7bb-n8crj 1/1 Running 0 7s
mongodb-8f797cf5b-qll9r 0/1 ContainerCreating 0 12s
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl get pods -l app=flask-todo-app
NAME READY STATUS RESTARTS AGE
flask-todo-app-78dbc4c7bb-mz6b8 1/1 Running 0 35s
flask-todo-app-78dbc4c7bb-n8crj 1/1 Running 0 35s
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl delete pod flask-todo-app-78dbc4c7bb-n8crj
pod "flask-todo-app-78dbc4c7bb-n8crj" deleted
^C
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl get pods -l app=flask-todo-app
NAME READY STATUS RESTARTS AGE
flask-todo-app-78dbc4c7bb-lm68d 1/1 Running 0 18s
flask-todo-app-78dbc4c7bb-mz6b8 1/1 Running 0 70s
flask-todo-app-78dbc4c7bb-n8crj 1/1 Terminating 0 70s
```

Scaling from 2 - 4 pods

```
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl scale deployment flask-todo-app --replicas=4
deployment.apps/flask-todo-app scaled
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl get pods -l app=flask-todo-app -w
NAME READY STATUS RESTARTS AGE
flask-todo-app-78dbc4c7bb-lm68d 1/1 Running 0 97s
flask-todo-app-78dbc4c7bb-mz6b8 1/1 Running 0 2m29s
flask-todo-app-78dbc4c7bb-ph7wh 1/1 Running 0 5s
flask-todo-app-78dbc4c7bb-tktxg 1/1 Running 0 5s
^C
```

Scaling down from 4 to 2

```
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl scale deployment flask-todo-app --replicas=2
deployment.apps/flask-todo-app scaled
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl get pods -l app=flask-todo-app -w
NAME READY STATUS RESTARTS AGE
flask-todo-app-78dbc4c7bb-lm68d 1/1 Running 0 2m15s
flask-todo-app-78dbc4c7bb-mz6b8 1/1 Running 0 3m7s
flask-todo-app-78dbc4c7bb-ph7wh 1/1 Terminating 0 43s
flask-todo-app-78dbc4c7bb-tktxg 1/1 Terminating 0 43s
^C
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl get pods -l app=flask-todo-app -w
NAME READY STATUS RESTARTS AGE
flask-todo-app-78dbc4c7bb-lm68d 1/1 Running 0 2m42s
flask-todo-app-78dbc4c7bb-mz6b8 1/1 Running 0 3m34s
```

Rollout update

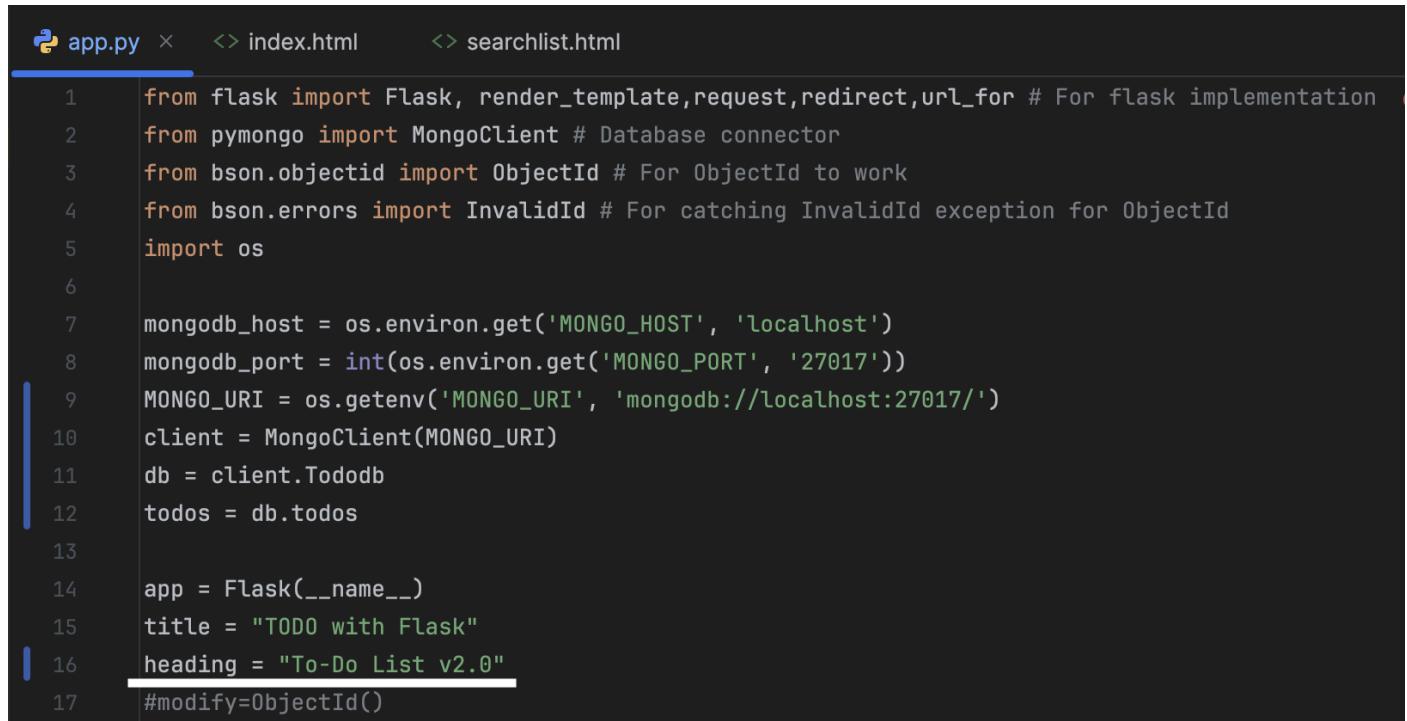
```
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl get pods -l app=flask-todo-app -w
NAME READY STATUS RESTARTS AGE
flask-todo-app-78dbc4c7bb-lm68d 1/1 Running 0 3h38m
flask-todo-app-78dbc4c7bb-mz6b8 1/1 Running 0 3h39m
flask-todo-app-78dbc4c7bb-lm68d 1/1 Terminating 0 3h39m
flask-todo-app-5b4845b7d4-m7lwk 0/1 Pending 0 0s
flask-todo-app-5b4845b7d4-m7lwk 0/1 Pending 0 0s
flask-todo-app-5b4845b7d4-m7lwk 0/1 ContainerCreating 0 0s
flask-todo-app-5b4845b7d4-74gwr 0/1 Pending 0 0s
flask-todo-app-5b4845b7d4-74gwr 0/1 Pending 0 0s
flask-todo-app-5b4845b7d4-74gwr 0/1 ContainerCreating 0 0s
flask-todo-app-5b4845b7d4-m7lwk 1/1 Running 0 2s
flask-todo-app-78dbc4c7bb-mz6b8 1/1 Terminating 0 3h40m
flask-todo-app-5b4845b7d4-74gwr 1/1 Running 0 2s
```

```
[base) yogarajalakshmi-s@Yogarajalakshmis-MacBook-Pro eks-development % kubectl ]  
apply -f flask-deployment-eks.yaml  
  
deployment.apps/flask-todo-app configured  
service/flask-todo-service unchanged  
(base) yogarajalakshmi-s@Yogarajalakshmis-MacBook-Pro eks-development % kubectl ]  
rollout status deployment/flask-todo-app  
  
deployment "flask-todo-app" successfully rolled out
```

Rollout history

```
(base) yogarajalakshmi-s@Yogarajalakshmis-MacBook-Pro eks-development % kubectl rollout history deployment/flask-todo-app  
  
deployment.apps/flask-todo-app  
REVISION  CHANGE-CAUSE  
1          <none>  
2          <none>
```

Updating text to see if the app is updated



The screenshot shows a code editor window with the file 'app.py' open. The code is a Flask application for a todo list. The file contains the following code:

```
 1  from flask import Flask, render_template,request,redirect,url_for # For flask implementation  
 2  from pymongo import MongoClient # Database connector  
 3  from bson.objectid import ObjectId # For ObjectId to work  
 4  from bson.errors import InvalidId # For catching InvalidId exception for ObjectId  
 5  import os  
 6  
 7  mongodb_host = os.environ.get('MONGO_HOST', 'localhost')  
 8  mongodb_port = int(os.environ.get('MONGO_PORT', '27017'))  
 9  MONGO_URI = os.getenv('MONGO_URI', 'mongodb://localhost:27017/')  
10  client = MongoClient(MONGO_URI)  
11  db = client.Tododb  
12  todos = db.todos  
13  
14  app = Flask(__name__)  
15  title = "TODO with Flask"  
16  heading = "To-Do List v2.0"  
17  #modify=ObjectId()
```

To-Do List v2.0

ALL	Uncompleted	Completed	About
No Tasks in the List !!			
Add a Task			
<input type="text" value="Taskname"/> <input type="text" value="Enter Description here..."/> <input type="text" value="mm/dd/yyyy"/> <input type="button" value=""/>			
<input type="button" value="Priority"/> <input type="button" value="Create"/>			

Rollback the change

```
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl rollout undo deployment/flask-todo-app
deployment.apps/flask-todo-app rolled back
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl get pods -l app=flask-todo-app -w
NAME          READY   STATUS      RESTARTS   AGE
flask-todo-app-5b4845b7d4-74gwr  1/1    Terminating   0          6m51s
flask-todo-app-5b4845b7d4-m71wk  1/1    Terminating   0          6m51s
flask-todo-app-78dbc4c7bb-b29pz  1/1    Running     0          5s
flask-todo-app-78dbc4c7bb-p9g27  1/1    Running     0          5s
[^C]
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl rollout history deployment/flask-todo-app
deployment.apps/flask-todo-app
REVISION  CHANGE-CAUSE
2        <none>
3        <none>
```

ToDo Reminder

ALL	Uncompleted	Completed	About
No Tasks in the List !!			
Add a Task			
<input type="text" value="Taskname"/> <input type="text" value="Enter Description here..."/> <input type="text" value="mm/dd/yyyy"/> <input type="button" value=""/>			
<input type="button" value="Priority"/> <input type="button" value="Create"/>			

Roll forward

To-Do List v2.0

ALL Uncompleted Completed About

No Tasks in the List !!

Add a Task

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

Create

flask-deployment-eks.yaml

```
# Rolling update strategy for Part 6
strategy:
  type: RollingUpdate
  rollingUpdate:
    maxUnavailable: 1 # Max 1 pod can be down during update
    maxSurge: 1         # Max 1 extra pod during update

  selector:
    matchLabels:
      app: flask-todo-app

template:
  metadata:
    labels:
      app: flask-todo-app
      version: v2 # For tracking versions during rolling updates
  spec:
    containers:
      - name: flask-todo-app

      # ⚠ CHANGE THIS to your teammate's Docker Hub image!
      image: yogas007/flask-todo-app:v2
```

Task 7 - Liveliness and Readiness probe

```

^C
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl describe pod -l app=flask-todo-app | grep -A 10 "Liveness:"
  Liveness: http-get http://:5000/ delay=30s timeout=5s period=15s #success=1 #failure=3
  Readiness: http-get http://:5000/ delay=10s timeout=5s period=10s #success=1 #failure=3
  Environment:
    MONGO_URI: mongodb://admin:password123@mongodb-service:27017/
    FLASK_ENV: production
  Mounts:
    /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-8ctqv (ro)
Conditions:
  Type        Status
  PodReadyToStartContainers  True
  Initialized  True
  -
  Liveness: http-get http://:5000/ delay=30s timeout=5s period=15s #success=1 #failure=3
  Readiness: http-get http://:5000/ delay=10s timeout=5s period=10s #success=1 #failure=3
  Environment:
    MONGO_URI: mongodb://admin:password123@mongodb-service:27017/
    FLASK_ENV: production
  Mounts:
    /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-xbtt (ro)
Conditions:
  Type        Status
  PodReadyToStartContainers  True
  Initialized  True
  Ready       True
  -
  Readiness: http-get http://:5000/ delay=10s timeout=5s period=10s #success=1 #failure=3
  Environment:
    MONGO_URI: mongodb://admin:password123@mongodb-service:27017/
    FLASK_ENV: production
  Mounts:
    /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-xbtt (ro)
Conditions:
  Type        Status
  PodReadyToStartContainers  True
  Initialized  True
  Ready       True
  -

```

Delete pod and see it recreating

```

^C
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl delete pod flask-todo-app-85fd7575fc-4mkqn
pod "flask-todo-app-85fd7575fc-4mkqn" deleted
[ ^C
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl get pods -l app=flask-todo-app -w
NAME                  READY   STATUS      RESTARTS   AGE
flask-todo-app-85fd7575fc-4mkqn  1/1     Terminating   0          7m26s
flask-todo-app-85fd7575fc-5djjn  1/1     Running     0          7m26s
flask-todo-app-85fd7575fc-kdx8t  1/1     Running     0          11s

(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl delete pod -l app=mongodb
pod "mongodb-8f797cf5b-6fk77" deleted
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl get pods -l app=flask-todo-app -w
NAME                  READY   STATUS      RESTARTS   AGE
flask-todo-app-85fd7575fc-7ph9w  1/1     Running     0          2m31s
flask-todo-app-85fd7575fc-lvr7c  1/1     Running     0          2m31s

```

Probe failures and restarts

```
[base) yogarajalakshmi-s@YogarajshmisMBP eks-development % kubectl get events --sort-by=.lastTimestamp | grep -i unhealthy
5m55s      Warning  Unhealthy          pod/flask-todo-app-85fd7575fc-5djjn  Liveness probe failed: Get "http://192.168.45.105:5000/": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
5m55s      Warning  Unhealthy          pod/flask-todo-app-85fd7575fc-kdx8t  Liveness probe failed: Get "http://192.168.22.162:5000/": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
5m5s       Warning  Unhealthy          pod/flask-todo-app-85fd7575fc-5djjn  Readiness probe failed: Get "http://192.168.45.105:5000/": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
5m       Warning  Unhealthy          pod/flask-todo-app-85fd7575fc-kdx8t  Readiness probe failed: Get "http://192.168.22.162:5000/": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
62s       Warning  Unhealthy          pod/flask-todo-app-85fd7575fc-7ph9w  Liveness probe failed: Get "http://192.168.1.98:5000/": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
62s       Warning  Unhealthy          pod/flask-todo-app-85fd7575fc-7ph9w  Readiness probe failed: Get "http://192.168.1.98:5000/": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
62s       Warning  Unhealthy          pod/flask-todo-app-85fd7575fc-lvr7c  Readiness probe failed: Get "http://192.168.55.189:5000/": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
62s       Warning  Unhealthy          pod/flask-todo-app-85fd7575fc-lvr7c  Liveness probe failed: Get "http://192.168.55.189:5000/": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
(base) yogarajalakshmi-s@YogarajshmisMBP eks-development %
```

Events:

Type	Reason	Age	From	Message
----	-----	---	----	-----
Normal	Scheduled	4m22s	default-scheduler	Successfully assigned default/flask-todo-app-85fd7575fc-lvr7c to ip-192-168-56-171.ec2.internal
Normal	Pulling	4m22s	kubelet	Pulling image "yogas007/flask-todo-app:v2"
Normal	Pulled	4m22s	kubelet	Successfully pulled image "yogas007/flask-todo-app:v2" in 151ms (151ms including waiting). Image size: 49787665 bytes.
Normal	Created	4m22s	kubelet	Created container: flask-todo-app
Normal	Started	4m22s	kubelet	Started container flask-todo-app
Warning	Unhealthy	107s (x2 over 3m37s)	kubelet	Readiness probe failed: Get "http://192.168.55.189:5000/": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
Warning	Unhealthy	107s	kubelet	Liveness probe failed: Get "http://192.168.55.189:5000/": context deadline exceeded (Client.Timeout exceeded while awaiting headers)

Prometheus Alerts Page

The screenshot shows the Prometheus Alerts page with the URL `3.92.142.133:30090/alerts?page=1`. The interface includes a top navigation bar with tabs for Prometheus, Query, Alerts (which is active), and Status. Below the navigation is a search bar with filters for rule group state and name/labels, and a pagination control showing page 1 of 2. The main content area displays the `alertmanager.rules` configuration file from `/etc/prometheus/rules/prometheus-prometheus-kube-prometheus-prometheus-rulefiles-0/monitoring-prometheus-kube-prometheus-alertmanager.rules-8fec20ce-bdb9-41c1-a214-e11168ae3ac3.yaml`. The configuration lists eight inactive alerts under the heading `INACTIVE (8)`:

- AlertmanagerFailedReload
- AlertmanagerMembersInconsistent
- AlertmanagerFailedToSendAlerts
- AlertmanagerClusterFailedToSendAlerts
- AlertmanagerConfigInconsistent
- AlertmanagerClusterDown

Slack Configuration

The screenshot shows the 'Incoming Webhooks' section of the Slack Configuration interface. On the left, a sidebar lists various app features like Socket Mode, Install App, Manage Distribution, and Features (App Home, Agents & AI Apps, Work Object Previews, Workflow Steps, Org Level Apps, Incoming Webhooks, Interactivity & Shortcuts, Slash Commands, Steps from Apps, OAuth & Permissions, Event Subscriptions, User ID Translation, App Manifest, Beta Features, and Submit to Slack Marketplace). The 'Incoming Webhooks' option is selected and highlighted in blue. The main content area provides instructions for using normal HTTP requests with a JSON payload to send messages. It also notes that a bot user will be added if none exists. A note about deactivating incoming webhooks is present. Below this, a section titled 'Webhook URLs for Your Workspace' explains how to dispatch messages using a curl command. A sample curl command is shown in a code block:

```
curl -X POST -H 'Content-type: application/json' --data '{"text":"Hello, World!"}' https://hooks.slack.com/services/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
```

A 'Copy' button is available to copy the URL. The table below lists the current webhook URLs.

Webhook URL	Channel	Added By

Slack Alerts

The screenshot shows the Slack workspace interface for the 'NYU-Cloud-Computing-Assignment' team. The left sidebar includes sections for Home, DMS, Activity, Files, and More. The 'More' section is expanded, showing Channels (# all-nyu-cloud-computing-assignment, # kubernetes-alerts, # new-channel, # social), Direct messages (Nishanth Kotla, Yoga you, Invite people), Apps (Slackbot, Kubernetes Alerts), and Add apps. The main area displays the '# kubernetes-alerts' channel. The channel header shows it is a default channel. A message from Slackbot at 21:50 says: 'This channel has been made a **default channel** for all new members who join **NYU-Cloud-Computing-Assignment**. If you want to remove this channel from the default channels, you can do so anytime in your [workspace admin settings](#)'. A message from Yoga at 21:51 says: 'added an integration to this channel: **Kubernetes Alerts**'. A message from Nishanth Kotla at 21:52 says: 'joined #kubernetes-alerts'. A message from 'Kubernetes Alerts APP' at 22:41 says: 'Test alert from Kubernetes - webhook is working!'. The message input field at the bottom contains the text: 'Message #kubernetes-alerts'.