SIT727 Cloud Automation Technologies

High Distinction Task 7.2HD - Project Delivery

Submitted By: Adeel Ahmed 224404186

"Micronews" – A Kubernetes-Based News Aggregation and Summarization System

Github: https://github.com/AdeelAhmedIqbal/Micronews-k8s

Panopto: https://deakin.au.panopto.com/Panopto/Pages/Viewer.aspx?id=98bff5be-303b-4385-808c-b2ed00cb6ce4&start=0

Youtube (unlisted): https://youtu.be/rQeSXegbIqg

Drive: https://drive.google.com/file/d/1B3_1PfX5VY5h4-mB0OgB5myedK1d3mMW/view?usp=sharing

Description of the Solution

I plan to build a simple platform called **Micronews** that gathers articles from public news sources and generates short summaries. The main goal is to give users a quick way to see what's happening without having to read every full article.

Why It's Interesting

Micronews highlights the power of microservices and shows how easy it is to scale parts of an application. For example, if there are more articles coming in than usual (perhaps due to a major news event), we can scale up the summarizer. Then, when things slow down, we can scale it back to save resources. This setup also demonstrates how different containers can work together seamlessly through Kubernetes.

MICRONEWS

Student: Adeel Ahmed | Student ID: 224404186

Microservices

AGGREGATOR: PULLS HEADLINES AND ARTICLE TEXT

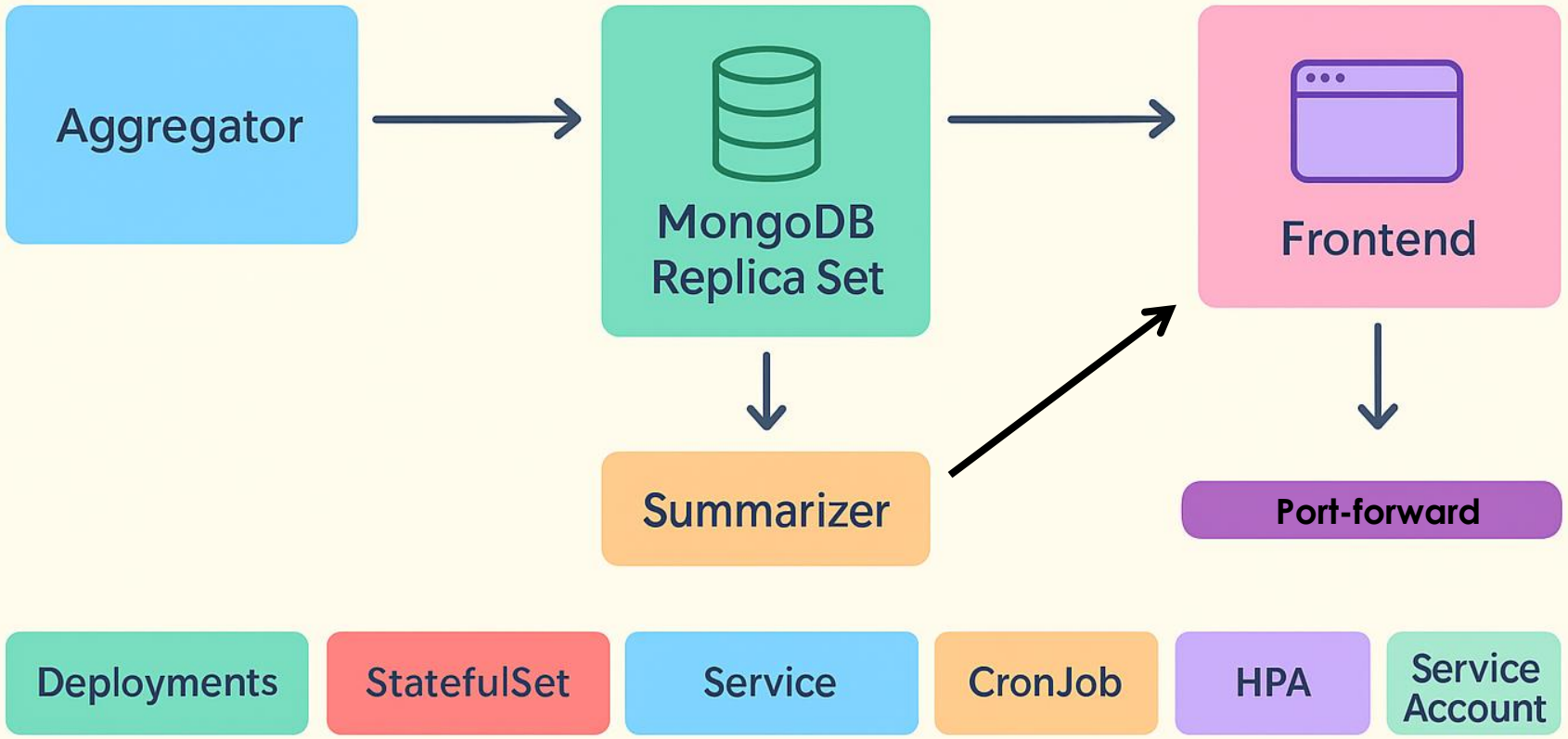
DATABASE: STORES ARTICLES RELIABLY

SUMMARIZER:

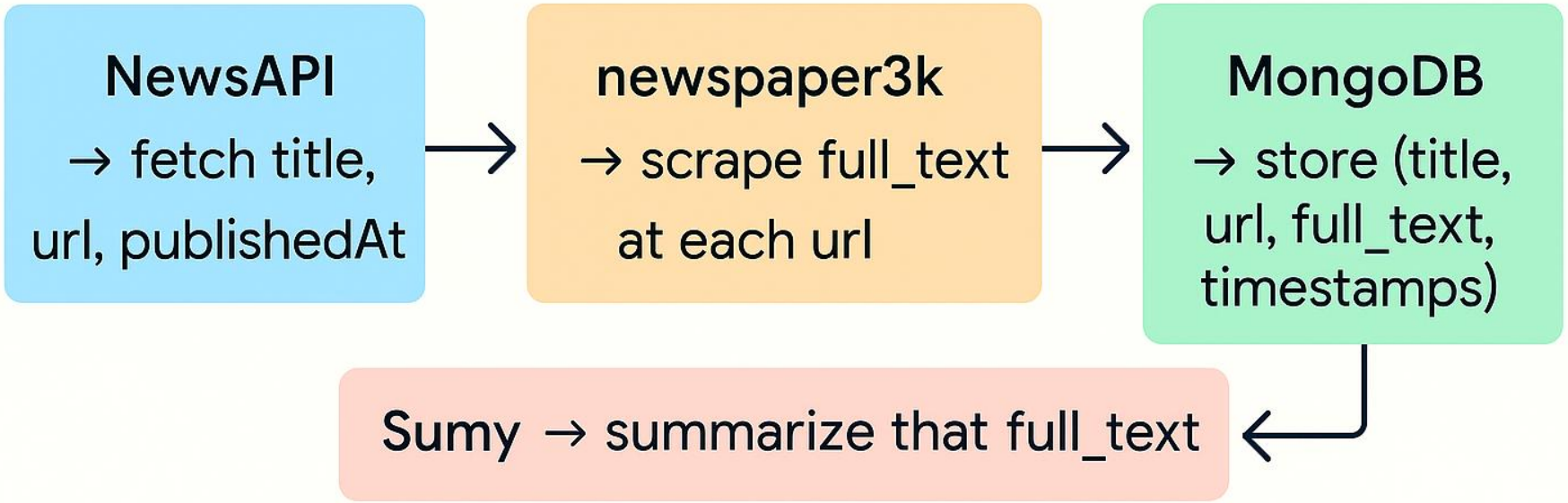
LEVERAGES NLP ALGORITHMS TO EXTRACT THE MOST IMPACTFUL SENTENCES FROM FULL-LENGTH ARTICLES

FRONTEND: PRESENTS CONTENT AND SUMMARIES TO USERS

Architecture



Architecture



Aggregator Overview

A containerized Python microservice that:

- Fetches full articles from NewsAPI's top-headlines endpoint
- Parses each story and stores only new entries in MongoDB
- Exposes a /fetch REST endpoint on port 8000 for downstream use
- Built with Flask, packaged in Docker, and deployed via a Kubernetes Deployment
- Automatically scales under load with a Horizontal Pod Autoscaler

Database Overview

- MongoDB replica set for high-availability
- Persistent storage across pod restarts
- Simple headless service for discovery

Summarizer Overview

- Stateless API for text summarization
- On-demand processing of stored articles
- Returns concise summaries via JSON

Frontend Overview

- Renders articles and summaries in a clean UI
- Configurable service URLs for internal calls

Core Kubernetes Components

- Deployments: manage service replicas
- StatefulSet: handle MongoDB durability
- Services: internal DNS for each component
- Secrets: secure API keys and URIs

Operational Enhancements

- Resource Limits & Requests for stability
- Horizontal Pod Autoscaler for scaling
- CronJob for regular backups
- ServiceAccounts & RBAC for minimal permissions

Monitoring & Recovery

- Metrics Server & Dashboard for live metrics and logs
- mongodump/mongorestore pattern for backup and restore

RBAC & Service Accounts

Each microservice runs under its own service account:

- frontend-sa
- aggregator-sa
- summarizer-sa

kubernetes > ! frontend-role.yaml

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: frontend-reader
  namespace: default
rules:
  - apiGroups: [""]
    resources: ["configmaps","secrets"]
    verbs:     ["get","list"]
```

kubernetes > ! frontend-rolebinding.yaml

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: bind-frontend-reader
  namespace: default
subjects:
  - kind: ServiceAccount
    name: frontend-sa
    namespace: default
roleRef:
  kind: Role
  name: frontend-reader
  apiGroup: rbac.authorization.k8s.io
```

adeel@Ubuntu-20: ~/Micronews-k8s/kubernetes

```
adeel@Ubuntu-20:~/Micronews-k8s/kubernetes$ kubectl auth can-i get configmaps --as=system:serviceaccount:default:frontend-sa
yes
adeel@Ubuntu-20:~/Micronews-k8s/kubernetes$ kubectl auth can-i list secrets --as=system:serviceaccount:default:frontend-sa
yes
adeel@Ubuntu-20:~/Micronews-k8s/kubernetes$ kubectl auth can-i list pods --as=system:serviceaccount:default:frontend-sa
no
adeel@Ubuntu-20:~/Micronews-k8s/kubernetes$ kubectl auth can-i create deployments --as=system:serviceaccount:default:frontend-sa
no
adeel@Ubuntu-20:~/Micronews-k8s/kubernetes$
```

Backup & Restore

- A Kubernetes CronJob runs every night to dump the entire micronews database into a timestamped, gzipped archive on a PersistentVolume

```
adeel@Ubuntu-20:~/Micronews-k8s$ k get cronjobs
NAME                  SCHEDULE    TIMEZONE   SUSPEND   ACTIVE   LAST SCHEDULE   AGE
mongodb-daily-backup  0 2 * * *   <none>     False     0        12h             13h
adeel@Ubuntu-20:~/Micronews-k8s$
```

- Archives on the host

- To restore directly into the live replica-set:

```
adeel@Ubuntu-20:~$ mongorestore \
>   --gzip \
>   --archive=2025-05-30.gz \
>   --uri="mongodb://mongo-0.mongo-service:27017,mongo-1.mongo-service:27017,mongo-2.mongo-service:27017/micronews?replicaSet=rs0" \
>   --drop
```

- To restore into a scratch DB for inspection:

```
adeel@Ubuntu-20:~$ mongorestore \
>   --gzip \
>   --archive=2025-05-30.gz \
>   --nsFrom="micronews.*" \
>   --nsTo="restoredb.*" \
>   --uri="mongodb://mongo-0.mongo-service:27017,mongo-1.mongo-service:27017,mongo-2.mongo-service:27017"
```

- If you need JSON output, export with:

```
adeel@Ubuntu-20:~$ mongoexport --db=restoredb --collection=articles --out=articles.json
```