

DevOps Final Assignment Report

Mohammad Raiyan Buhiyan Loreen

1. Project Overview

This report summarizes the implementation of the DevOps final assignment through the development, containerization, CI/CD automation, and Kubernetes deployment of a web application named **DevBlog**. The project demonstrates a complete DevOps lifecycle from development to deployment using industry-standard tools and platforms such as GitHub Actions, Docker, and Kubernetes (k0s).

2. Tools and Technologies Used

- **Frontend Framework:** React
- **Containerization:** Docker
- **CI/CD:** GitHub Actions
- **Kubernetes Platform:** k0s
- **Source Control:** GitHub

3. How to Run the Project

Locally without Docker

```
npm install
npm start
```

Using Docker

```
docker build -t devopsblog .
docker run -p 5000:5000 devopsblog
```

Pull from Docker Hub

```
docker pull raiyan77/devopsblog
docker run -p 5000:5000 raiyan77/devopsblog
```

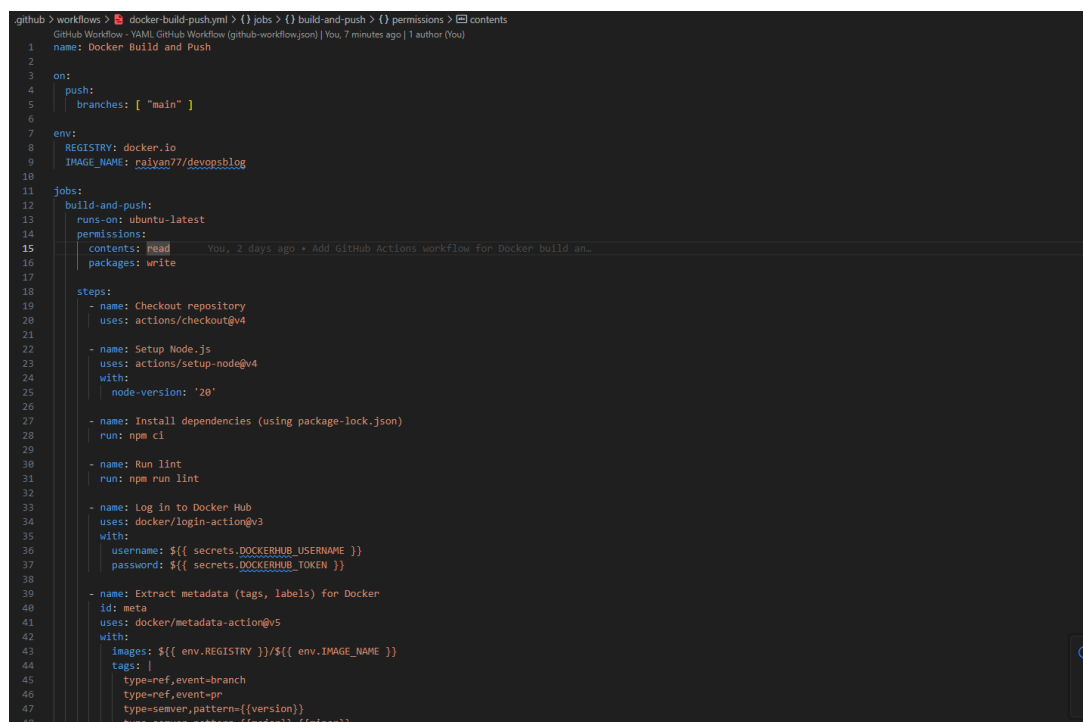
4. GitHub and Docker Hub Links

- GitHub Repository: <https://github.com/RAIYANBHUIYAN/DevOpsBlog.git>
- Docker Hub Image: <https://hub.docker.com/r/raian77/devopsblog>

5. CI/CD Workflow Explanation

The CI/CD pipeline is defined in the GitHub Actions workflow file located at `.github/workflows/ci.yml`.
The workflow:

- Triggers on every push to the main branch
- Installs dependencies and builds the React app
- Builds a Docker image
- Optionally pushes the Docker image to Docker Hub



```
github > workflows > docker-build-push.yml > {} jobs > {} build-and-push > {} permissions > {} contents
GitHub Workflow - YAML GitHub Workflow (github-workflow.json) | You, 7 minutes ago | 1 author (You)
1 name: Docker Build and Push
2
3 on:
4   push:
5     branches: [ "main" ]
6
7 env:
8   REGISTRY: docker.io
9   IMAGE_NAME: raiyan77/devopsblog
10
11 jobs:
12   build-and-push:
13     runs-on: ubuntu-latest
14     permissions:
15       contents: read
16       packages: write
17
18     steps:
19       - name: Checkout repository
20         uses: actions/checkout@v4
21
22       - name: Setup Node.js
23         uses: actions/setup-node@v4
24         with:
25           node-version: '20'
26
27       - name: Install dependencies (using package-lock.json)
28         run: npm ci
29
30       - name: Run lint
31         run: npm run lint
32
33       - name: Log in to Docker Hub
34         uses: docker/login-action@v3
35         with:
36           username: ${{ secrets.DOCKERHUB_USERNAME }}
37           password: ${{ secrets.DOCKERHUB_TOKEN }}
38
39       - name: Extract metadata (tags, labels) for Docker
40         id: meta
41         uses: docker/metadata-action@v5
42         with:
43           images: ${{ env.REGISTRY }}/${{ env.IMAGE_NAME }}
44           tags: |
45             type=ref,event=branch
46             type=ref,event=pr
47             type=semver,pattern={{version}}
48             type=sha,format={{sha}}
49
```

Figure 1: GitHub Actions Workflow File (Part 1)

```

11 jobs:
12   build-and-push:
13     permissions:
14       packages: write
15     steps:
16       - name: Checkout repository
17         uses: actions/checkout@v4
18       - name: Setup Node.js
19         uses: actions/setup-node@v4
20         with:
21           node-version: '20'
22       - name: Install dependencies (using package-lock.json)
23         run: npm ci
24       - name: Run lint
25         run: npm run lint
26       - name: Log in to Docker Hub
27         uses: docker/login-action@v3
28         with:
29           username: ${{ secrets.DOCKERHUB_USERNAME }}
30           password: ${{ secrets.DOCKERHUB_TOKEN }}
31       - name: Extract metadata (tags, labels) for Docker
32         id: meta
33         uses: docker/metadata-action@v5
34         with:
35           images: ${{ env.REGISTRY }}/${{ env.IMAGE_NAME }}
36           tags: |
37             type=ref,event=branch
38             type=ref,event=pr
39             type=semver,pattern={{version}}
40             type=semver,pattern={{major}}.{{minor}}
41             type=sha,format=short
42       - name: Build and push Docker image
43         uses: docker/build-push-action@v5
44         with:
45           context: .
46           push: true
47           tags: ${{ steps.meta.outputs.tags }}
48           labels: ${{ steps.meta.outputs.labels }}

```

Figure 2: GitHub Actions Workflow File (Part 2)

6. Screenshots

Project Structure

Running Website

Docker Configuration

Docker Hub Image

GitHub Action Successful Run

Kubernetes Deployment

7. Learnings and Challenges

What I Learned:

- Setting up a CI pipeline using GitHub Actions
- Containerizing a web app with Docker
- Creating Kubernetes manifests and deploying locally using k0s
- Managing cloud-ready infrastructure in a DevOps environment

Challenges Faced:

- Initially configuring GitHub Actions with React builds

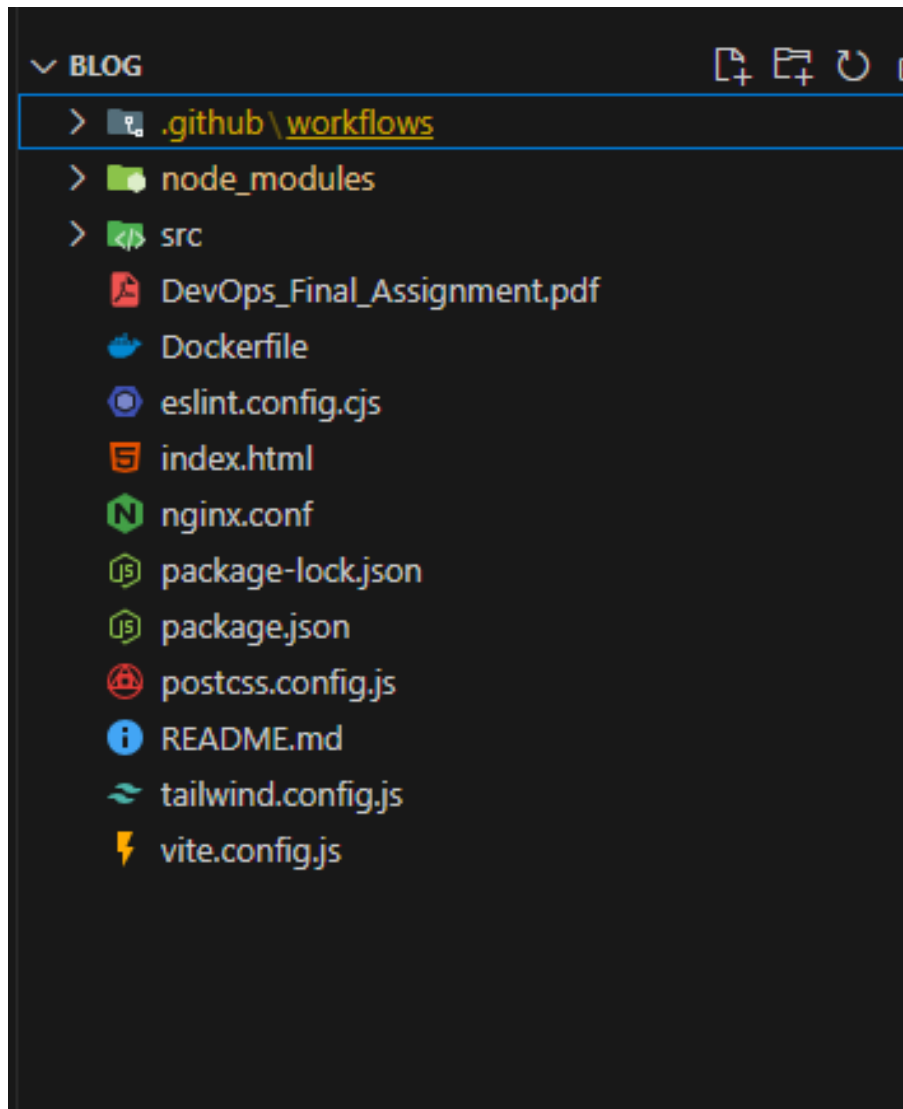


Figure 3: Project Folder Structure

- Debugging Dockerfile base image compatibility
- Managing Kubernetes port exposure and verifying the deployment status

Potential Improvements:

- Integrate unit tests for automation in CI pipeline
- Add monitoring and logging with Prometheus/Grafana
- Set up production-level deployment on cloud Kubernetes clusters

8. Video Explanation

A complete video explanation of this project can be found here: <https://youtu.be/13NoAvofQ88>

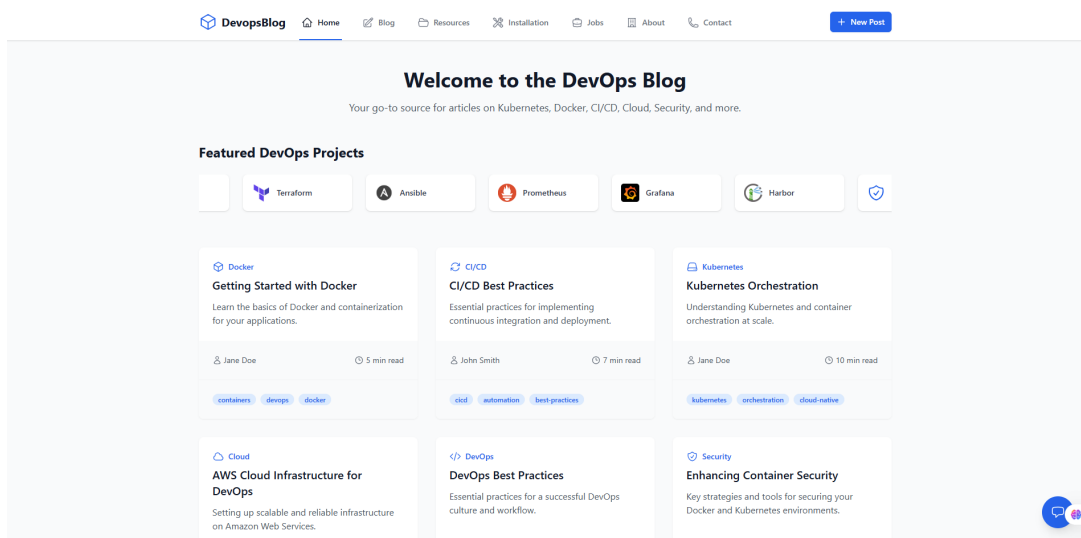


Figure 4: Blog Homepage

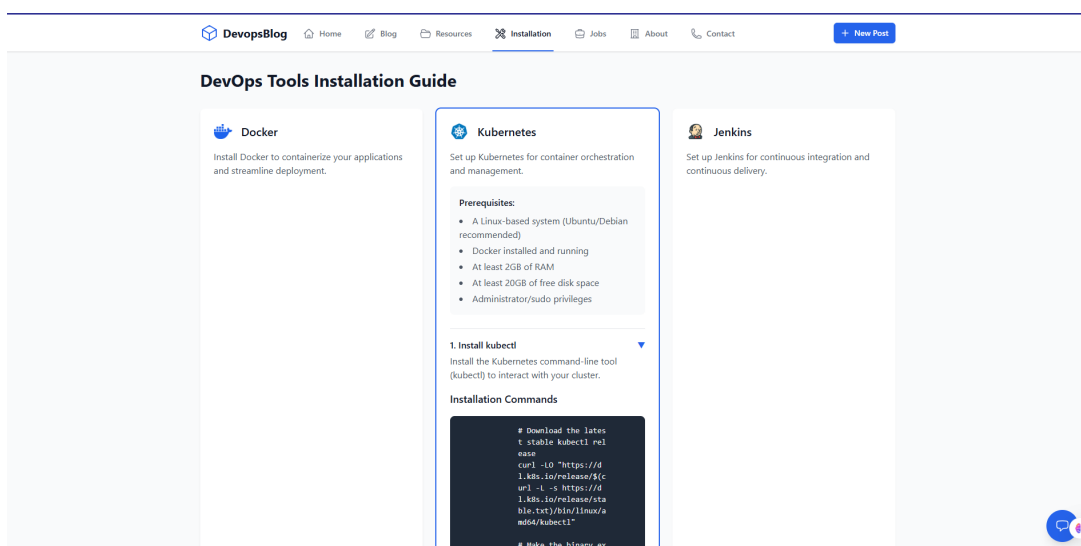


Figure 5: Blog Post Section

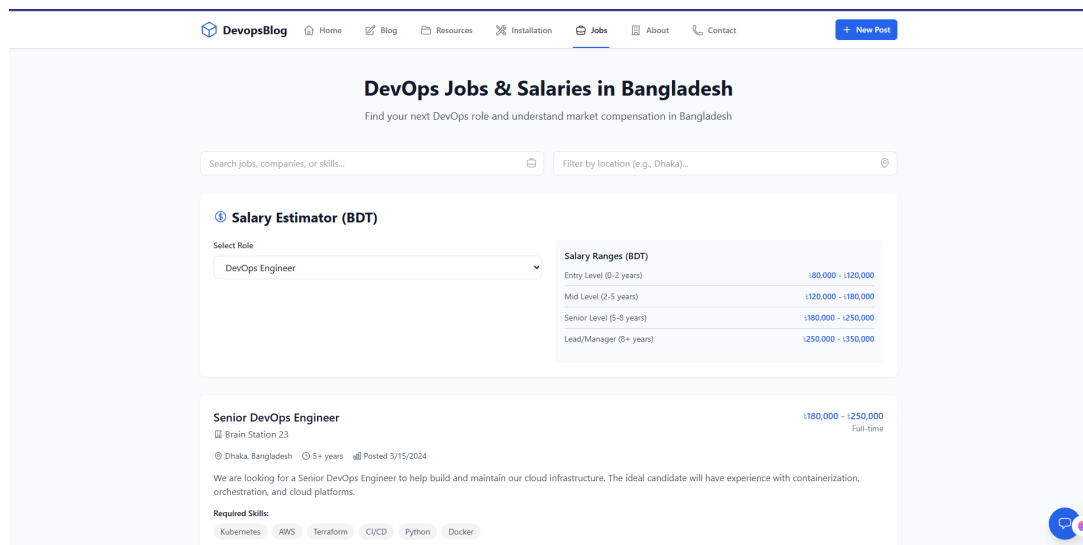


Figure 6: Additional Content View

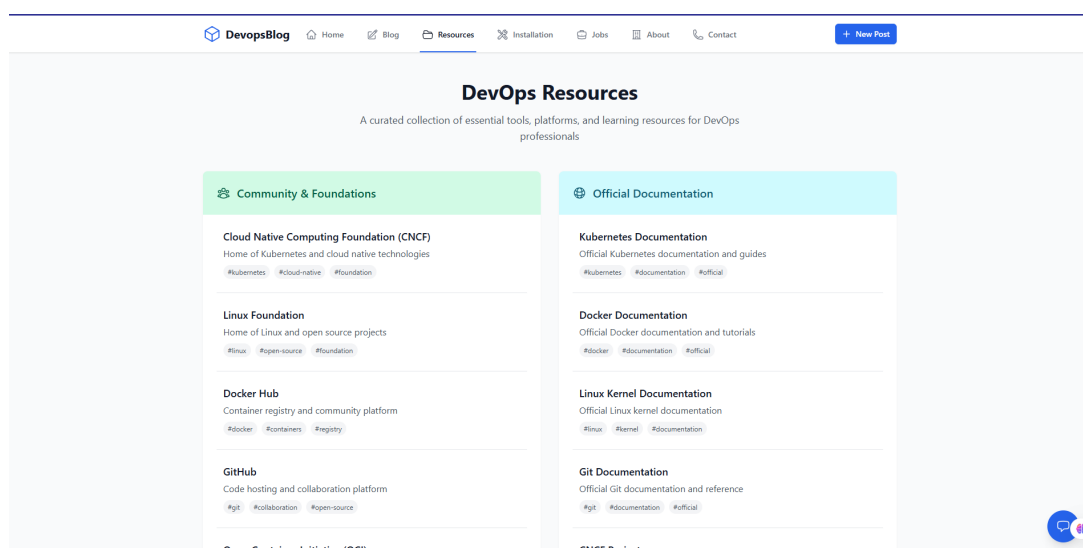


Figure 7: Mobile Responsiveness

```


1 # Build stage
2 FROM node:20-alpine AS build
3     You, 3 days ago • initial commit ...
4 WORKDIR /app
5
6 # Install build dependencies
7 RUN apk add --no-cache python3 make g++
8
9 # Copy package files
10 COPY package*.json ./
11
12 # Clean install dependencies with proper platform support
13 RUN npm cache clean --force && \
14     rm -rf node_modules package-lock.json && \
15     npm install
16
17 # Copy source code
18 COPY . .
19
20 # Set proper permissions
21 RUN chown -R node:node /app
22
23 # Switch to non-root user
24 USER node
25
26 # Build the application using local vite
27 RUN npm run build
28
29 # Production stage
30 FROM nginx:alpine
31
32 # Copy built assets from build stage
33 COPY --from=build /app/dist /usr/share/nginx/html
34
35 # Copy nginx configuration
36 COPY nginx.conf /etc/nginx/conf.d/default.conf
37
38 # Expose port 5000
39 EXPOSE 5000
40
41 # Start nginx
42 CMD ["nginx", "-g", "daemon off;"]

```


Figure 8: Dockerfile


Repositories / devopsblog / General

Using 0 of 1 private repositories. [Get more](#)


raiyan77/devopsblog 

Last pushed 14 minutes ago · Repository size: 20.2 MB





[Add a description](#) 

[Add a category](#) 


[General](#) [Tags](#) [Image Management](#) [BETA](#) [Collaborators](#) [Webhooks](#) [Settings](#)

Tags  [Activate](#)

This repository contains 0 tag(s).

Tag	OS	Type	Pulled	Pushed
 sha-9b8c228		Image	less than 1 day	14 minutes
 main		Image	less than 1 day	14 minutes

[See all](#)



Build with
Docker Build Cloud

Accelerate image build times with access to cloud-based builders and shared cache.

Docker Build Cloud executes builds on optimally-dimensioned cloud infrastructure with dedicated per-organization isolation.

Get faster builds through shared caching across your team, native multi-platform support, and encrypted data transfer - all without managing infrastructure.

[Go to Docker Build Cloud](#) →

Figure 9: Docker Hub Repository

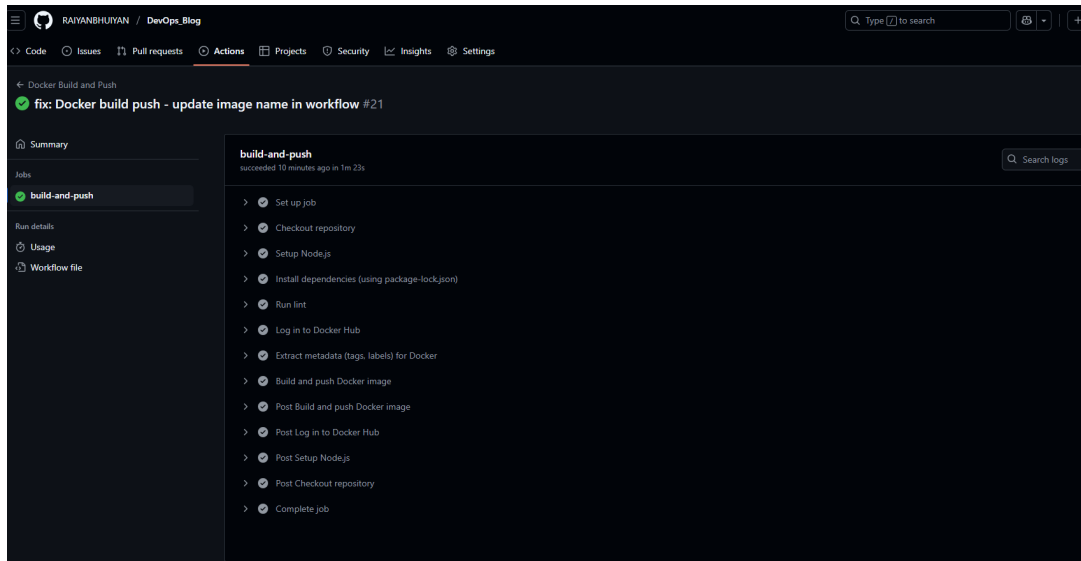


Figure 10: Successful GitHub Actions Run

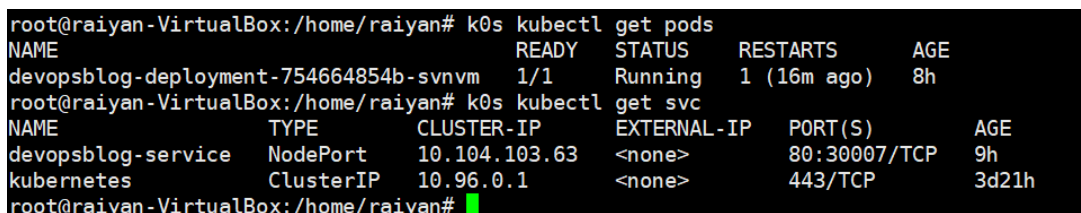


Figure 11: k0s Pods and Service Running

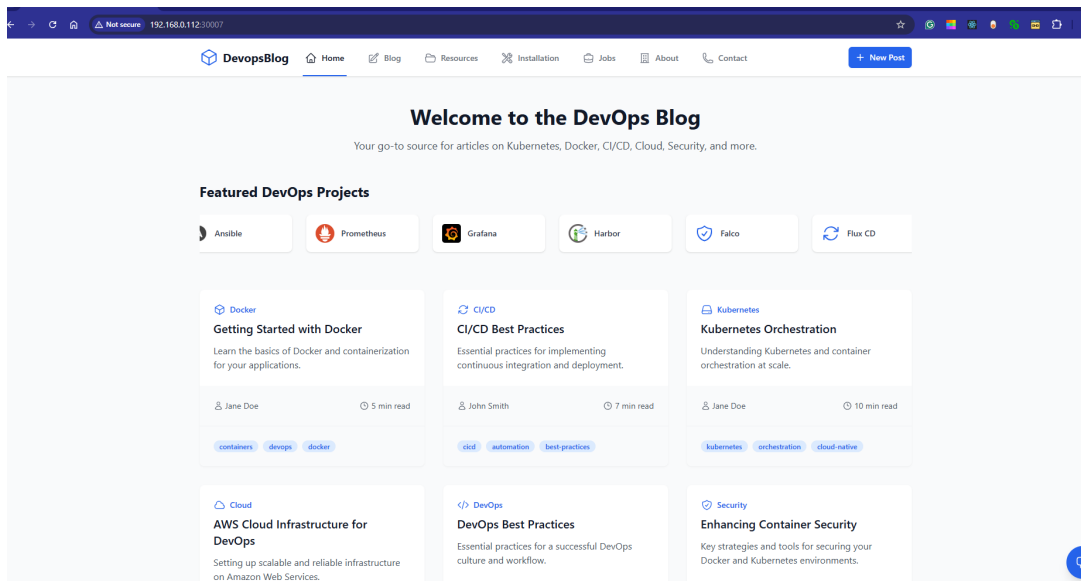


Figure 12: Application Running via k0s

Name	Namespace	Container	CPU	Memory	Restart	Controlled By	Node	QoS	Age	Status
aws-node-lgdmz	kube-system		0.000	61.2MiB	0	DaemonSet	ip-10-230-1	Burstable	2d13h	Running
aws-node-qgqnm	kube-system		0.000	65.6MiB	0	DaemonSet	ip-10-230-1	Burstable	7h23m	Running
aws-node-t9s97	kube-system		0.000	61.3MiB	0	DaemonSet	ip-10-230-1	Burstable	7h20m	Running
cluster-autoscaler-aws-clust	kube-system		0.000	37.9MiB	0	ReplicaSet	ip-10-230-1	Burstable	2d13h	Running
coredns-54588cf8c-f86kp	kube-system		0.000	14.3MiB	0	ReplicaSet	ip-10-230-1	Burstable	7h19m	Running
coredns-54588cf8c-xqfp8	kube-system		0.000	14.8MiB	0	ReplicaSet	ip-10-230-1	Burstable	2d13h	Running
ebs-csi-controller-7c7db7bb	kube-system		0.000	57.6MiB	0	ReplicaSet	ip-10-230-1	Burstable	2d13h	Running
ebs-csi-controller-7c7db7bb	kube-system		0.000	48.3MiB	0	ReplicaSet	ip-10-230-1	Burstable	7h19m	Running
ebs-csi-node-2xx9m	kube-system		0.000	19.5MiB	0	DaemonSet	ip-10-230-1	Burstable	2d13h	Running
ebs-csi-node-58msm	kube-system		0.000	18.7MiB	0	DaemonSet	ip-10-230-1	Burstable	7h20m	Running
ebs-csi-node-n4742	kube-system		0.000	18.9MiB	0	DaemonSet	ip-10-230-1	Burstable	7h23m	Running
eks-pod-identity-agent-567ls	kube-system		0.000	5.8MiB	0	DaemonSet	ip-10-230-1	BestEffort	7h20m	Running
eks-pod-identity-agent-v2jtz	kube-system		0.000	6.1MiB	0	DaemonSet	ip-10-230-1	BestEffort	7h23m	Running
eks-pod-identity-agent-zwfj4	kube-system		0.000	5.9MiB	0	DaemonSet	ip-10-230-1	BestEffort	2d13h	Running
kube-proxy-2bzq2	kube-system		0.000	12.7MiB	0	DaemonSet	ip-10-230-1	Burstable	2d13h	Running
kube-proxy-w6kcl	kube-system		0.000	12.5MiB	0	DaemonSet	ip-10-230-1	Burstable	7h20m	Running
kube-proxy-zk45r	kube-system		0.000	12.7MiB	0	DaemonSet	ip-10-230-1	Burstable	7h23m	Running
metrics-server-6486766f4f-h	kube-system		0.010	25.7MiB	0	ReplicaSet	ip-10-230-1	Burstable	2d13h	Running

Figure 13: Pods Visualization using Lens