

# My Contribution and Progress at Gyzertech

CI/CD IMPLEMENTATION | KUBERNETES | MONITORING

PRESENTED BY: FESTUS OKAGBARE

DATE: 1/07/2025



# Introduction

- ▶ Joined as DevOps Engineer
- ▶ Focused on automation, CI/CD, containerization, and monitoring

# What I Was Tasked With

- ▶ Implement CI/CD pipelines
- ▶ Containerize applications (app1, app2)
- ▶ Deploy to Azure VM and Kubernetes
- ▶ Set up monitoring (New Relic)

# What I Have Done So Far

- ▶ CI/CD pipelines for app1 (Dev & Staging), app2 (Dev & Staging)
- ▶ Dockerized applications
- ▶ Deployed to Azure VM and Kubernetes Cluster
- ▶ Set up monitoring using New Relic

# Screenshots – What's Working

- ▶ CI/CD pipeline run status (Success)
- ▶ Docker container running on Azure VM
- ▶ Kubernetes pods active (kubectl get pods)
- ▶ New Relic Dashboard showing live data

# App1 development branch CICD successful deployment

The screenshot displays the GitHub Actions interface for the 'glix-repo' repository. The main view shows the 'CI/CD Pipeline for Azure VM (Development)' workflow, which is currently running successfully. The workflow is triggered by a push to the 'development' branch by user 'seunAwonugba' at commit '5427700'. The status is 'Success' with a total duration of '5m 52s'. The workflow consists of three main stages: 'Matrix: build-development' (7 jobs completed), 'Matrix: db-migrations' (7 jobs completed), and 'deploy-development' (2m 11s). The left sidebar lists the jobs in the pipeline, including 'build-development' for various services and 'db-migrations' for different databases. The bottom of the interface shows options to view 'Run details', 'Usage', and the 'Workflow file'.

← CI/CD Pipeline for Azure VM (Development)

bug fix #131

build-development (billing\_service, ...)

build-development (certificate\_serv...

build-development (client\_service, cl...

build-development (employee\_serv...

build-development (server\_auth\_ser...

build-development (services\_server, ...)

build-development (settings\_server, ...)

db-migrations (auth, AUTH\_SERVER\_...

db-migrations (billing, BILLING\_SER...

db-migrations (certificate, CERTIFICA...

db-migrations (employee, EMPLOYE...

db-migrations (services, SERVICES\_S...

db-migrations (settings, SETTINGS\_S...

db-migrations (client, CLIENT\_SERVE...

deploy-development

Run details

Usage

Workflow file

Triggered via push last month

seunAwonugba pushed · 5427700 development

Status: Success

Total duration: 5m 52s

Artifacts: —

deployment.yaml

on: push

Matrix: build-development

7 jobs completed

Show all jobs

Matrix: db-migrations

7 jobs completed

Show all jobs

deploy-development 2m 11s

# App1 staging branch CI/CD successful

The screenshot displays the GitHub Actions interface for the 'glix-repo' repository. The main heading is 'Merge pull request #102 from GyzerTech/staging2 #94'. Below this, a table shows the pipeline's execution details:

Triggered via	Status	Total duration	Artifacts
Festiveokagbare pushed → c02cf35 staging	Success	9m 33s	—

The pipeline is defined in 'staging.yml' and runs on 'push'. The workflow consists of three main stages:

- Matrix: build-staging**: 7 jobs completed. (Show all jobs)
- Matrix: db-migrations**: 7 jobs completed. (Show all jobs)
- deploy-staging**: 5m 30s.

The left sidebar lists the jobs for each stage, all marked with green checkmarks indicating success. The bottom of the interface shows options for 'Run details', 'Usage', and 'Workflow file'.

# App2 Development branch CI/CD successful deployment

The screenshot shows a GitHub Actions workflow run for the repository 'GyzerTech / schedu'. The workflow is named 'CI/CD Pipeline for Azure VM (Development)' and the specific run is 'fix host profile id #556'. The run status is 'Success' and it took '6m 13s' to complete. The workflow was triggered by a push to the 'development' branch by user 'seunAwonugba'.

**Jobs:**

- build-development (billing, billing, billi...
- build-development (schedu\_backend, ...
- build-development (schedu\_event, sch...
- build-development (workflow, workflo...
- db-migrations (auth, DEV\_DB\_AUTH, A...
- db-migrations (billing, BILLING\_DEV\_D...
- db-migrations (schedu\_event, EVENT\_...
- db-migrations (workflow, WORKFLOW...
- deploy-development

**deploy.yaml workflow diagram:**

```
graph LR; A["Matrix: build-development  
4 jobs completed  
Show all jobs"] --> B["Matrix: db-migrations  
4 jobs completed  
Show all jobs"]; B --> C["deploy-development  
2m 41s"]
```

**Run details:**

- Usage
- Workflow file



# App2 Staging branch CI/CD successful deployment to AKS

The screenshot displays a GitHub Actions workflow run for the repository 'GyzerTech / schedu'. The workflow is titled 'updated ingress file #89' and has a status of 'Success'. The run was triggered via a push last week by user 'Festiveokagbare' with commit 'cf99ce3' to the 'festus-staging2' branch. The total duration of the run is 7m 37s.

The workflow is named 'aks-staging.yml' and is triggered on a push. The jobs are as follows:

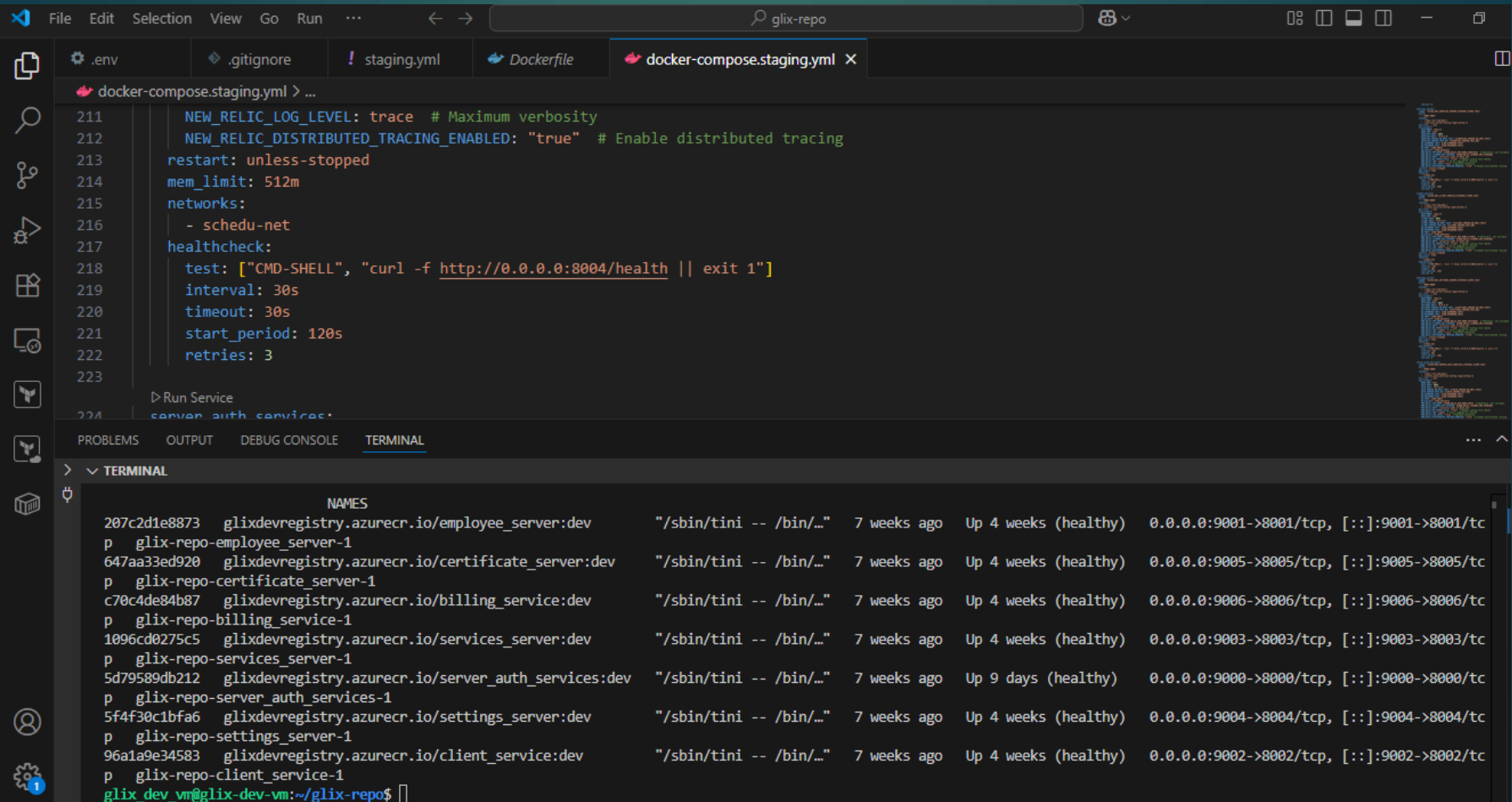
- Build & Push Images (billing, billi...)
- Build & Push Images (schedu\_bac...)
- Build & Push Images (schedu\_eve...)
- Build & Push Images (workflow, ...)
- Run DB Migrations (billing, k8s/b...)
- Run DB Migrations (schedu-back...)
- Run DB Migrations (schedu-event...)
- Run DB Migrations (workflow, k8s...)
- Deploy to AKS

The workflow steps are:

- Matrix: Build & Push Images (4 jobs completed)
- Matrix Run DB Migrations (4 jobs completed)
- Deploy to AKS (3m 44s)

The workflow is successful, and the deployment to AKS is complete.

# Containers running for App1 Dev Azure VM



The screenshot shows the Visual Studio Code editor with the `docker-compose.staging.yml` file open. The file contains configuration for a service named `glix-repo`, including environment variables, restart policy, memory limit, network, and healthcheck. The terminal at the bottom displays the output of the `docker-compose up` command, showing that several services have been successfully started and are in a healthy state.

```
211     NEW_RELIC_LOG_LEVEL: trace # Maximum verbosity
212     NEW_RELIC_DISTRIBUTED_TRACING_ENABLED: "true" # Enable distributed tracing
213 restart: unless-stopped
214 mem_limit: 512m
215 networks:
216   - schedu-net
217 healthcheck:
218   test: ["CMD-SHELL", "curl -f http://0.0.0.0:8004/health || exit 1"]
219   interval: 30s
220   timeout: 30s
221   start_period: 120s
222   retries: 3
223
224 > Run Service
225 server_auth_services:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

TERMINAL

```
207c2d1e8873 glixdevregistry.azurecr.io/employee_server:dev "/sbin/tini -- /bin/..." 7 weeks ago Up 4 weeks (healthy) 0.0.0.0:9001->8001/tcp, [::]:9001->8001/tc
p glix-repo-employee_server-1
647aa33ed920 glixdevregistry.azurecr.io/certificate_server:dev "/sbin/tini -- /bin/..." 7 weeks ago Up 4 weeks (healthy) 0.0.0.0:9005->8005/tcp, [::]:9005->8005/tc
p glix-repo-certificate_server-1
c70c4de84b87 glixdevregistry.azurecr.io/billing_service:dev "/sbin/tini -- /bin/..." 7 weeks ago Up 4 weeks (healthy) 0.0.0.0:9006->8006/tcp, [::]:9006->8006/tc
p glix-repo-billing_service-1
1096cd0275c5 glixdevregistry.azurecr.io/services_server:dev "/sbin/tini -- /bin/..." 7 weeks ago Up 4 weeks (healthy) 0.0.0.0:9003->8003/tcp, [::]:9003->8003/tc
p glix-repo-services_server-1
5d79589db212 glixdevregistry.azurecr.io/server_auth_services:dev "/sbin/tini -- /bin/..." 7 weeks ago Up 9 days (healthy) 0.0.0.0:9000->8000/tcp, [::]:9000->8000/tc
p glix-repo-server_auth_services-1
5f4f30c1bfa6 glixdevregistry.azurecr.io/settings_server:dev "/sbin/tini -- /bin/..." 7 weeks ago Up 4 weeks (healthy) 0.0.0.0:9004->8004/tcp, [::]:9004->8004/tc
p glix-repo-settings_server-1
96a1a9e34583 glixdevregistry.azurecr.io/client_service:dev "/sbin/tini -- /bin/..." 7 weeks ago Up 4 weeks (healthy) 0.0.0.0:9002->8002/tcp, [::]:9002->8002/tc
p glix-repo-client_service-1
glix_dev_vm@glix-dev-vm:~/glix-repo$
```

# Containers running for App1 Staging Azure VM

```
.env .gitignore staging.yml Dockerfile docker-compose.staging.yml X
docker-compose.staging.yml > ...
241 DO_DIALECT: ${DO_DIALECT}
242 NEW_RELIC_APP_NAME: ${NEW_RELIC_APP_NAME_AUTH} # Reference .env variable
243 NEW_RELIC_LICENSE_KEY_STAGING: ${NEW_RELIC_LICENSE_KEY_STAGING}
244 NEW_RELIC_HOST: collector.eu01.nr-data.net
245 NEW_RELIC_NO_CONFIG_FILE: "true" # Bypass config file checks
246 NEW_RELIC_LOG: stdout # Force logging to console
247 NEW_RELIC_LOG_LEVEL: trace # Maximum verbosity
248 NEW_RELIC_DISTRIBUTED_TRACING_ENABLED: "true" # Enable distributed tracing
249 restart: unless-stopped
250 mem_limit: 512m
251 networks:
252   - schedu-net
253 healthcheck:
254   test: ["CMD-SHELL", "curl -f http://0.0.0.0:8000/health || exit 1"]
255   interval: 30s
```

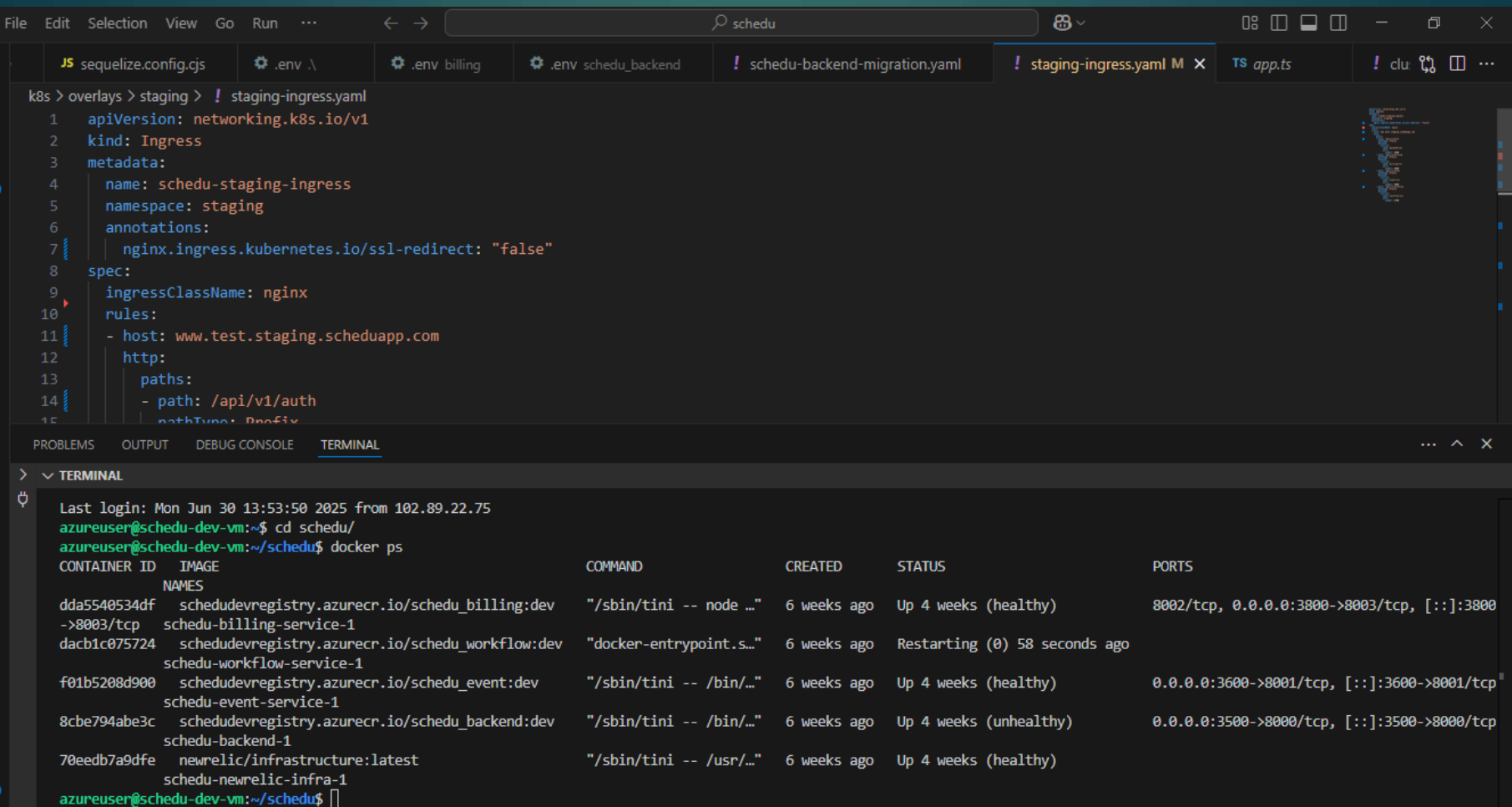
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

> ▼ TERMINAL

```

NAMES
0da73ae93bc7 glxstagingregistry.azurecr.io/settings_server:staging "/sbin/tini -- node ..." 7 weeks ago Up 4 weeks (healthy) 0.0.0.0:9004->8004/tcp, [::]:9004-
>8004/tcp glx-repo-settings_server-1
bc5a634ed7aa glxstagingregistry.azurecr.io/employee_server:staging "/sbin/tini -- node ..." 7 weeks ago Up 4 weeks (healthy) 0.0.0.0:9001->8001/tcp, [::]:9001-
>8001/tcp glx-repo-employee_server-1
5251b0dd0369 glxstagingregistry.azurecr.io/services_server:staging "/sbin/tini -- node ..." 7 weeks ago Up 4 weeks (healthy) 0.0.0.0:9003->8003/tcp, [::]:9003-
>8003/tcp glx-repo-services_server-1
efa8dd4b8c68 glxstagingregistry.azurecr.io/certificate_server:staging "/sbin/tini -- node ..." 7 weeks ago Up 4 weeks (healthy) 0.0.0.0:9005->8005/tcp, [::]:9005-
>8005/tcp glx-repo-certificate_server-1
0b7cb48d501e glxstagingregistry.azurecr.io/client_service:staging "/sbin/tini -- node ..." 7 weeks ago Up 4 weeks (healthy) 0.0.0.0:9002->8002/tcp, [::]:9002-
>8002/tcp glx-repo-client_service-1
baffbee97ada glxstagingregistry.azurecr.io/billing_service:staging "/sbin/tini -- node ..." 7 weeks ago Up 4 weeks (healthy) 0.0.0.0:9006->8006/tcp, [::]:9006-
>8006/tcp glx-repo-billing_service-1
b1abde0a5c9b glxstagingregistry.azurecr.io/server_auth_services:staging "/sbin/tini -- node ..." 7 weeks ago Up 4 weeks (healthy) 0.0.0.0:9000->8000/tcp, [::]:9000-
>8000/tcp glx-repo-server_auth_services-1
azureuser@glix-staging-vm:~/glix-repo$
```

# Containers running for App2 Dev Azure VM



The screenshot shows a Visual Studio Code editor with a Kubernetes manifest file named `staging-ingress.yaml` open. The manifest defines an Ingress resource for staging-ingress in the staging namespace, using the nginx ingress controller. It specifies a rule for `www.test.staging.scheduapp.com` with a path `/api/v1/auth`.

Below the editor, the TERMINAL window shows the output of the `docker ps` command, displaying a table of running containers on the `azureuser@schedu-dev-vm` machine.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
dda5540534df	schedudevregistry.azurecr.io/schedu_billing:dev	"/sbin/tini -- node ..."	6 weeks ago	Up 4 weeks (healthy)	8002/tcp, 0.0.0.0:3800->8003/tcp, [::]:3800
dacb1c075724	schedudevregistry.azurecr.io/schedu_workflow:dev	"docker-entrypoint.s..."	6 weeks ago	Restarting (0) 58 seconds ago	
f01b5208d900	schedudevregistry.azurecr.io/schedu_event:dev	"/sbin/tini -- /bin/..."	6 weeks ago	Up 4 weeks (healthy)	0.0.0.0:3600->8001/tcp, [::]:3600->8001/tcp
8cbe794abe3c	schedudevregistry.azurecr.io/schedu_backend:dev	"/sbin/tini -- /bin/..."	6 weeks ago	Up 4 weeks (unhealthy)	0.0.0.0:3500->8000/tcp, [::]:3500->8000/tcp
70eedb7a9dfe	newrelic/infrastructure:latest	"/sbin/tini -- /usr/..."	6 weeks ago	Up 4 weeks (healthy)	

The terminal prompt is `azureuser@schedu-dev-vm:~/schedu$`.

# Apps Pods running on Kubernetes with 2 replicas each and db-migrations

```
MINGW64:/c/Users/USER/Desktop/DEVOPS/shedu-Repo/schedu
USER@Festive MINGW64 ~/Desktop/DEVOPS/shedu-Repo/schedu (festus-staging2)
$ kubectl get pods -n staging
```

NAME	READY	STATUS	RESTARTS	AGE
backend-service-6f876c7c55-lbc9v	1/1	Running	0	6d13h
backend-service-6f876c7c55-mb9st	1/1	Running	0	6d13h
billing-69f4677d45-rhtjc	1/1	Running	0	5d15h
billing-69f4677d45-vm76s	1/1	Running	0	5d15h
billing-migration-job-nlgtg	0/1	Completed	0	5d15h
billing-service-78f9c756bb-84prt	1/1	Running	0	6d13h
billing-service-78f9c756bb-98x7f	1/1	Running	0	6d13h
cm-acme-http-solver-4vhb2	1/1	Running	0	5d15h
event-service-5d7f468b9f-28kdr	1/1	Running	0	6d13h
event-service-5d7f468b9f-jzdjq	1/1	Running	0	6d13h
health-fix-5b884fdbcb9-rnv6s	1/1	Running	0	4d15h
health-fix-5b884fdbcb9-s8xs5	1/1	Running	0	4d15h
health-fix-5b884fdbcb9-wfgmn	1/1	Running	0	4d15h
schedu-backend-5bdb589c88-bq2sw	1/1	Running	0	5d15h
schedu-backend-5bdb589c88-fwcr5	1/1	Running	0	5d15h
schedu-backend-migration-job-nwldh	0/1	Completed	0	5d15h
schedu-event-7564dd4bd8-n9vw4	1/1	Running	0	5d15h
schedu-event-7564dd4bd8-nmthr	1/1	Running	0	5d15h
schedu-event-migration-job-fls2f	0/1	Completed	0	5d15h
workflow-66c746fd9b-77hp5	1/1	Running	0	5d15h
workflow-66c746fd9b-zgqp6	1/1	Running	0	5d15h
workflow-migration-job-n56d9	0/1	Completed	0	5d15h
workflow-service-848dbdf65f-h6wtf	1/1	Running	0	6d13h
workflow-service-848dbdf65f-wvbjg	1/1	Running	0	6d13h

```
USER@Festive MINGW64 ~/Desktop/DEVOPS/shedu-Repo/schedu (festus-staging2)
$
```

# App1 staging-aks cluster infrastructure provision with Terraform on Azure portal

The screenshot displays the Microsoft Azure portal interface. At the top, the header includes the Microsoft Azure logo, a search bar, and a Copilot button. The user's profile, 'festus.okagbare@gyzer...', is visible in the top right corner. The main content area shows the 'staging-aks' Kubernetes service overview. A left-hand navigation pane lists various services like Activity log, Access control (IAM), Tags, Monitor, and more. The main pane is divided into sections: 'Essentials' with a table of cluster details, and 'Properties' with sub-sections for 'Kubernetes services' and 'Networking'.

**Microsoft Azure** Search resources, services, and docs (G+/)

Home > **staging-aks** Kubernetes service

Search

+ Create Connect Start Stop Delete Refresh Open in mobile Give feedback

**Essentials** JSON View

Resource group	: <a href="#">schedu-staging-resource-group</a>	Kubernetes version	: 1.31.7
Power state	: Running	API server address	: staging-aks-dns-mw8g4q0z.hcp.norwayeast.azmk...
Cluster operation st...	: Succeeded	Network configurat...	: <a href="#">Azure CNI Node Subnet</a>
Subscription	: <a href="#">Microsoft Azure Sponsorship</a>	Node pools	: 1 node pool
Location	: Norway East	Container registries	: <a href="#">Attach a registry</a>
Subscription ID	: 2ea23776-8393-446d-b086-3d60cdb02f8b		
Fleet Manager	: <a href="#">Click here to assign</a>		
Tags ( <a href="#">edit</a> )	: Environment : staging ManagedBy : Terraform		

Get started **Properties** Monitoring Recommendations

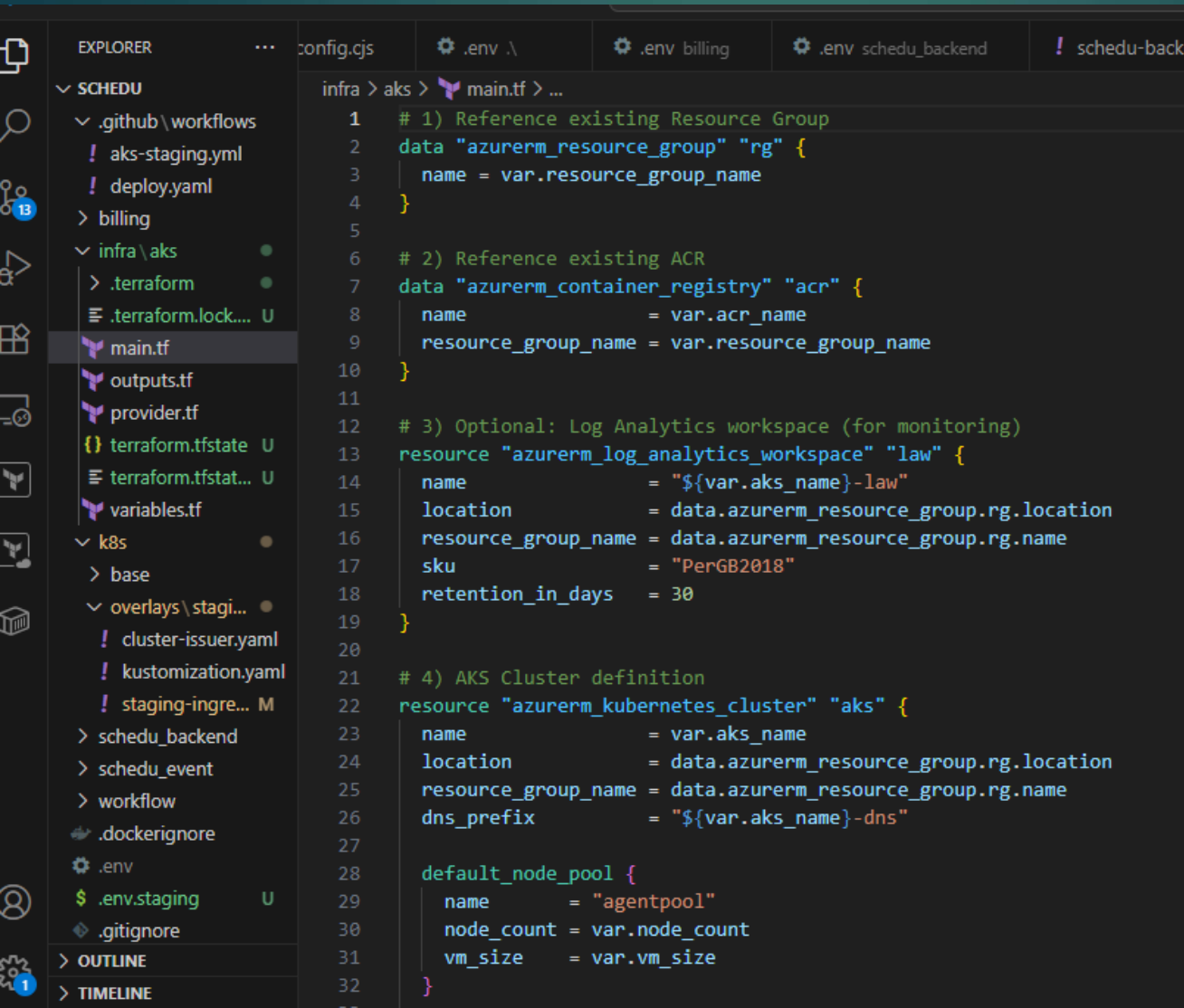
**Kubernetes services**

Encryption type	Encryption at-rest with a platform-managed key
Virtual node pools	Not enabled

**Networking**

API server address	staging-aks-dns-mw8g4q0z.hcp.norwayeast.azmk8s.io
Network configuration	Azure CNI Node Subnet

# Terraform code used for provisioning aks cluster



```
infra > aks > main.tf > ...
1  # 1) Reference existing Resource Group
2  data "azurerm_resource_group" "rg" {
3    name = var.resource_group_name
4  }
5
6  # 2) Reference existing ACR
7  data "azurerm_container_registry" "acr" {
8    name = var.acr_name
9    resource_group_name = var.resource_group_name
10 }
11
12 # 3) Optional: Log Analytics workspace (for monitoring)
13 resource "azurerm_log_analytics_workspace" "law" {
14   name = "${var.aks_name}-law"
15   location = data.azurerm_resource_group.rg.location
16   resource_group_name = data.azurerm_resource_group.rg.name
17   sku = "PerGB2018"
18   retention_in_days = 30
19 }
20
21 # 4) AKS Cluster definition
22 resource "azurerm_kubernetes_cluster" "aks" {
23   name = var.aks_name
24   location = data.azurerm_resource_group.rg.location
25   resource_group_name = data.azurerm_resource_group.rg.name
26   dns_prefix = "${var.aks_name}-dns"
27
28   default_node_pool {
29     name = "agentpool"
30     node_count = var.node_count
31     vm_size = var.vm_size
32   }
33 }
```



# Dbeaver dashboard showing a successful database migration from the cicd deployment pipeline

The screenshot displays the Dbeaver database management tool interface. The left sidebar shows a tree view of the database structure for 'auth\_db\_testfest' on 'schedu-staging-db.postgres.database.azure.com:5432'. The 'SequelizeMeta' table is selected, showing its schema and size (16K). The main window displays the 'SequelizeMeta' table data in a grid view, listing 17 records of migration scripts. The bottom status bar shows 'Project - General' and 'DataSource'.

id	name
1	20240925161438-create-availability.js
2	20240925161438-create-event.js
3	20240925161438-create-meeting-poll-event.js
4	20240925161438-create-meeting-poll-votes.js
5	20240925161438-create-meeting-poll.js
6	20240925161438-create-ticket.js
7	20240925161438-create-unavailability.js
8	20241011164422-create-event-availability.js
9	20241022152349-create-calendar.js
10	20241029101811-create-booking.js
11	20241128134635-create-meeting.js
12	20250219113533-create-event-log.js
13	20250219123424-create-analytics.js
14	20250312142443-create-advanced-settings.js
15	20250312144908-create-last-advanced-settings.js
16	20250326012101-create-event-questions.js
17	20250326012159-create-event-answers.js



# Successful docker Image push to app1 dev registry

← ↻ 🔒 https://portal.azure.com/#@gyzertech.com/resource/subscriptions/2ea23776-8393-446d-b086-3d60cdb02f8b/resourceGroups/glix\_dev\_resource\_group...

Microsoft Azure Search resources, services, and docs (G+/) Copilot

Home > Container registries > glixdevregistry

## Container registries

Gyzer Technologies Limited (gyzertech.com)

+ Create ... Group by none

*You are viewing a new version of Browse experience. Some features may be missing. [Click here to access the old experience.](#)*

<input type="checkbox"/>	Name ↑
<input type="checkbox"/>	glixdevregistry
<input type="checkbox"/>	glixstagingregistry
<input type="checkbox"/>	schedudevregistry
<input type="checkbox"/>	schedustageregistry

Showing 1 - 4 of 4. Display count: 10

### glixdevregistry | Repositories

Container registry

Search

Refresh Manage Deleted Repositories

Search to filter repositories ...

Repositories ↑↓	Cache Rule
<a href="#">billing_service</a>	
<a href="#">certificate_server</a>	
<a href="#">client_service</a>	
<a href="#">employee_server</a>	
<a href="#">server_auth_services</a>	
<a href="#">services_server</a>	
<a href="#">settings_server</a>	

- Access control (IAM)
- Tags
- Quick start
- Resource visualizer
- Events
- Settings
- Services
- Repositories**
- Webhooks
- Geo-replications
- Tasks
- Connected registries
- Cache
- Repository permissions

Add or remove favorites by pressing Ctrl+Shift+F

# Successful docker Image push to app1 staging registry

← ↻ 🔒 <https://portal.azure.com/#@gyzertech.com/resource/subscriptions/2ea23776-8393-446d-b086-3d60cdb02f8b/resourceGroups/glix-staging-resource-gr...> 🗒️ 🔊 ⭐ ⚙️ ? 👤 festus.okagbare@ GYZER TECHNOLOGIES

☰ Microsoft Azure 🔍 Search resources, services, and docs (G+/)

Home > Container registries > glixstagingregistry

## Container registries

Gyzer Technologies Limited (gyzertech.com)

+ Create ⚙️ Manage view ▾ ...

**Information** You are viewing a new version of Browse experience. Some features may be missing. [Click here to access the old experience.](#)

<input type="checkbox"/>	Name ↑
<input type="checkbox"/>	glixdevregistry
<input type="checkbox"/>	glixstagingregistry
<input type="checkbox"/>	schedudevregistry
<input type="checkbox"/>	schedustageregistry

Showing 1 - 4 of 4. Display  count: ▾

## glixstagingregistry | Repositories

Container registry

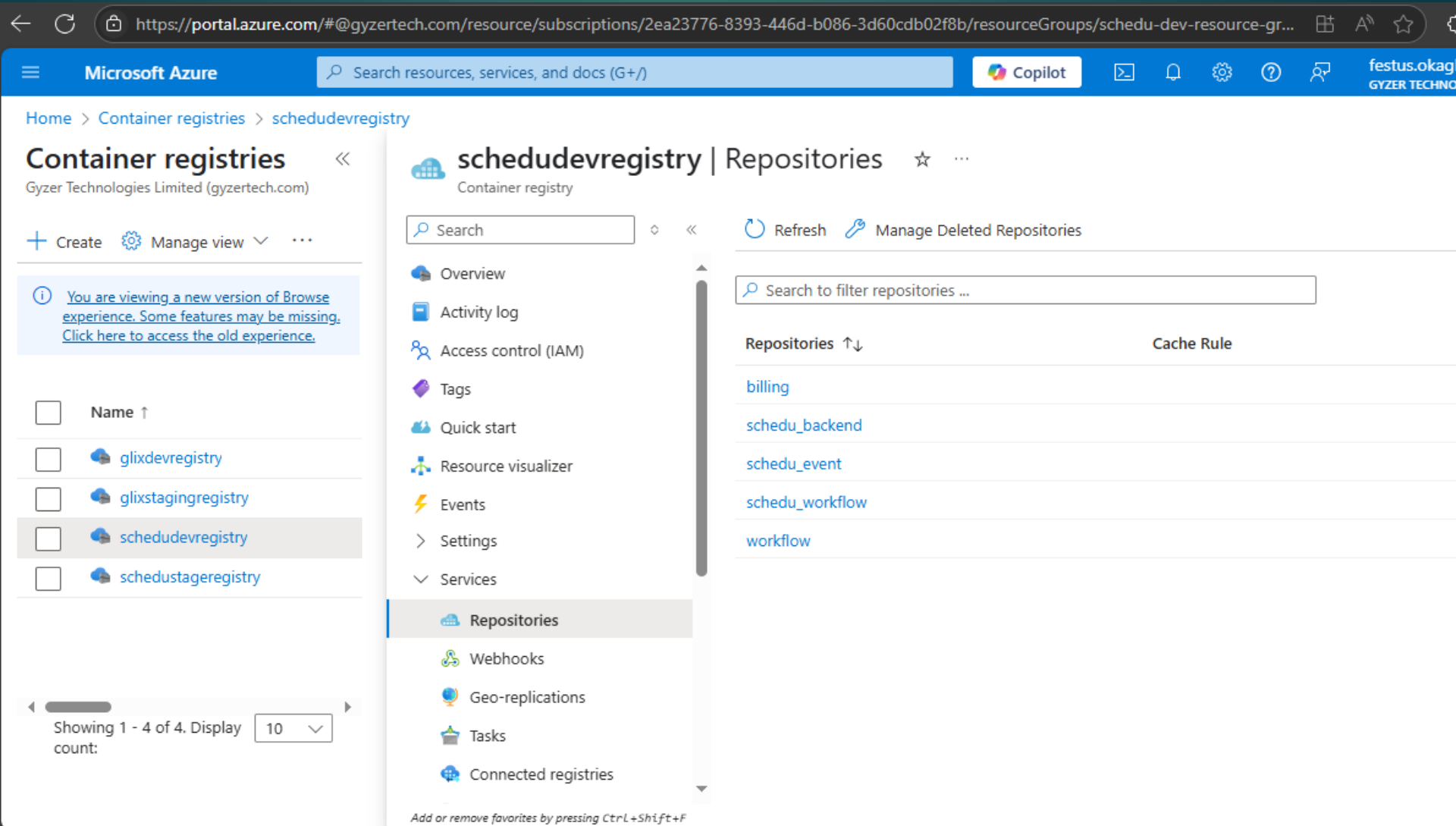
🔍 Search 🔁 Refresh 🔑 Manage Deleted Repositories

- Overview
- Activity log
- Access control (IAM)
- Tags
- Quick start
- Resource visualizer
- Events
- Settings
- Services
- Repositories**
- Webhooks
- Geo-replications
- Tasks
- Connected registries

🔍 Search to filter repositories ...

Repositories ↑↓	Cache Rule
<a href="#">billing_service</a>	
<a href="#">certificate_server</a>	
<a href="#">client_service</a>	
<a href="#">employee_server</a>	
<a href="#">server_auth_services</a>	
<a href="#">services_server</a>	
<a href="#">settings_server</a>	

5



# Successful docker Image push to app2 staging registry

The screenshot displays the Microsoft Azure portal interface for managing container registries. The top navigation bar includes the Microsoft Azure logo, a search bar, and the Copilot icon. The user's email, festus.okagbare@gyzert..., is visible in the top right corner.

The main content area is titled "Container registries" and shows a list of registries for GYZER TECHNOLOGIES LIMITED. The "schedustageregistry" is selected, and its "Repositories" page is displayed. The left sidebar contains a navigation menu with options like Overview, Activity log, Access control (IAM), Tags, Quick start, Resource visualizer, Events, Settings, Services, Repositories (selected), Webhooks, Geo-replications, Tasks, and Connected registries.

The "Repositories" page shows a list of repositories with columns for "Repositories" and "Cache Rule". The repositories listed are:

Repositories	Cache Rule
billing	
schedu_backend	
schedu_billing	
schedu_event	
schedu_workflow	
workflow	

At the bottom of the interface, a message states: "Add or remove favorites by pressing Ctrl+Shift+F".

# Tools I Worked With

- ▶ Docker
- ▶ GitHub Repos
- ▶ GitHub Actions
- ▶ Azure VM
- ▶ Kubernetes
- ▶ New Relic

# Self-Assessment

- ▶ Delivered key DevOps results
- ▶ Successfully **built and deployed multiple CI/CD pipelines** and containerized applications
- ▶ Deployed infrastructure for **Kubernetes orchestration and monitoring**
- ▶ Estimated contribution and performance: 80%+

# Conclusion & Appreciation

**THANK  
YOU!**