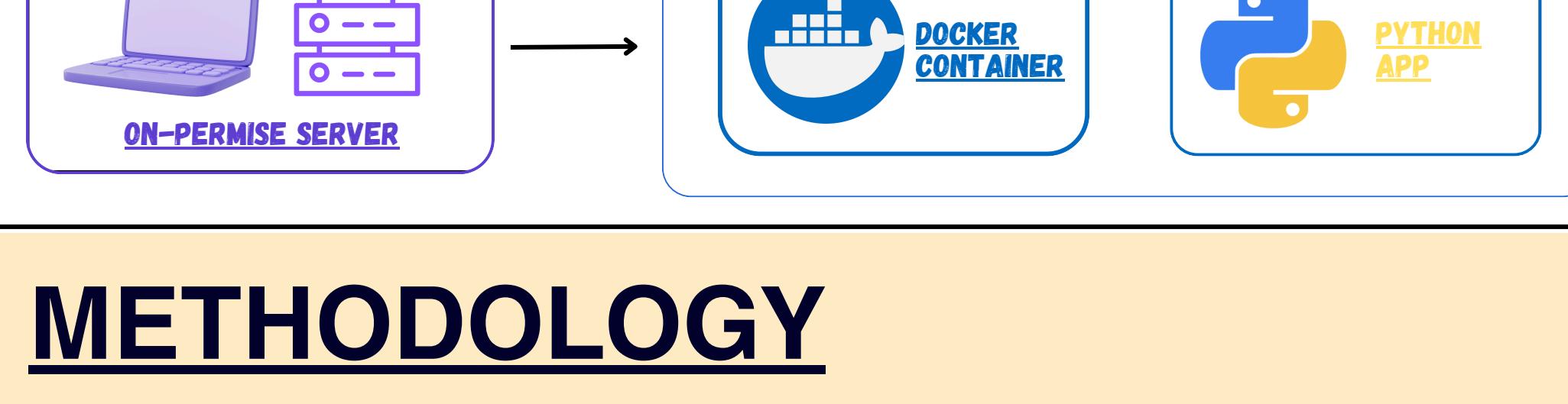


INTRODUCTION

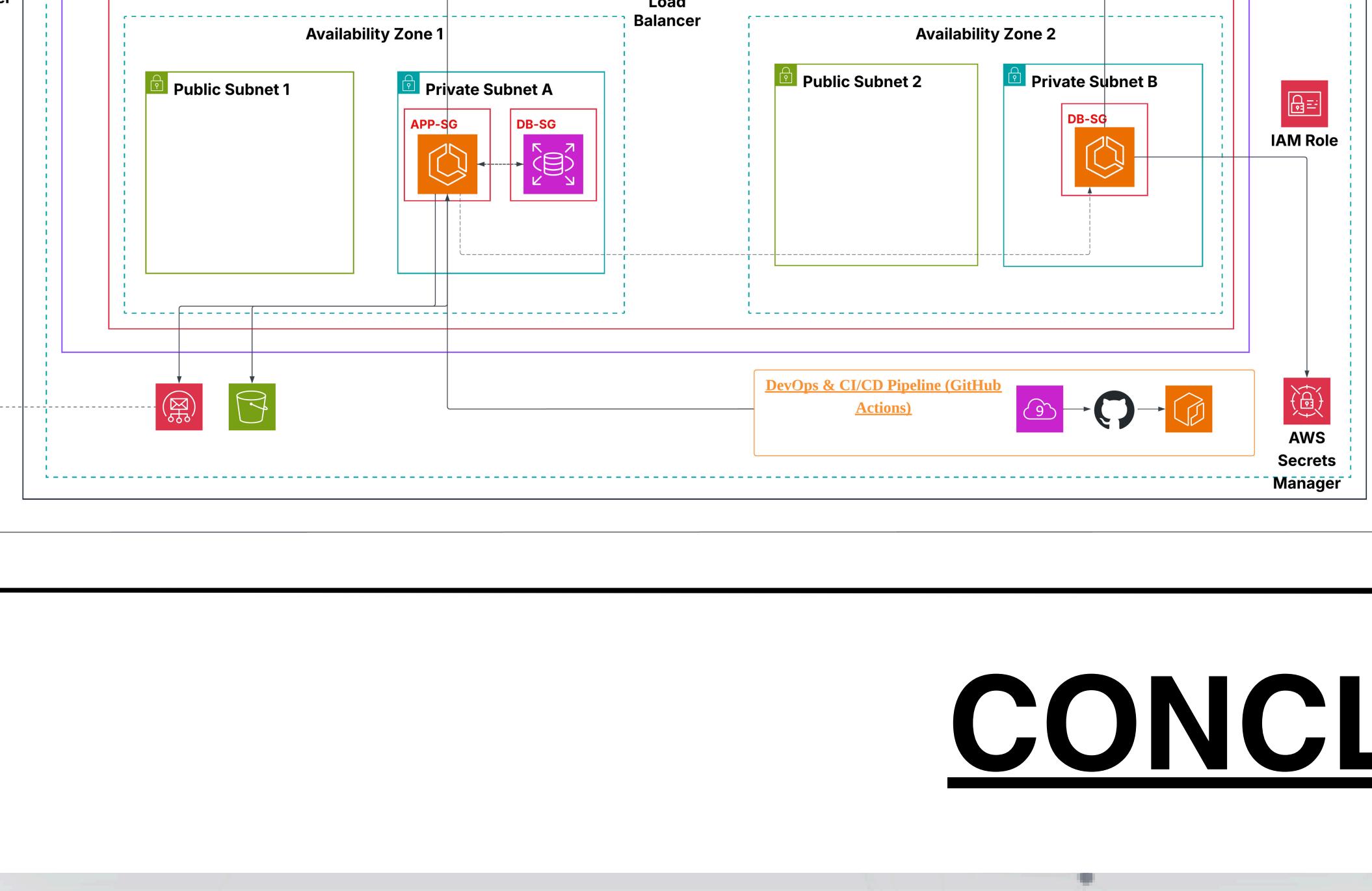
- Developed a containerized web application to modernize student record management and room allocation.
- Migrates legacy monolithic architecture to AWS Serverless (Fargate) for high availability.
- Solves issues of ephemeral storage in containers by integrating cloud-native persistence.



METHODOLOGY

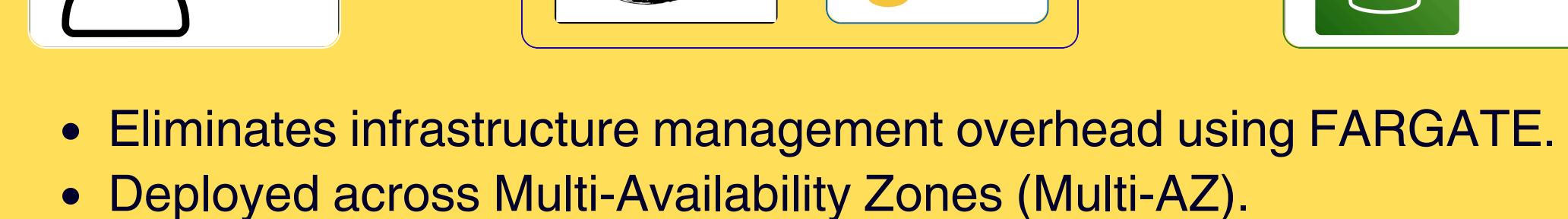
- Adopted a serverless approach using AWS Fargate to decouple the application from physical infrastructure, ensuring high availability and auto-scaling.
- Designed a custom VPC with strict subnet isolation. The Database resides in a private network, accessible only via the application container, protecting it from external threats.
- Implemented a modern GitOps workflow. Code changes trigger an automated pipeline (GitHub Actions) that builds, tests, and deploys Docker images with zero downtime.

ARCHITECTURE DIAGRAM



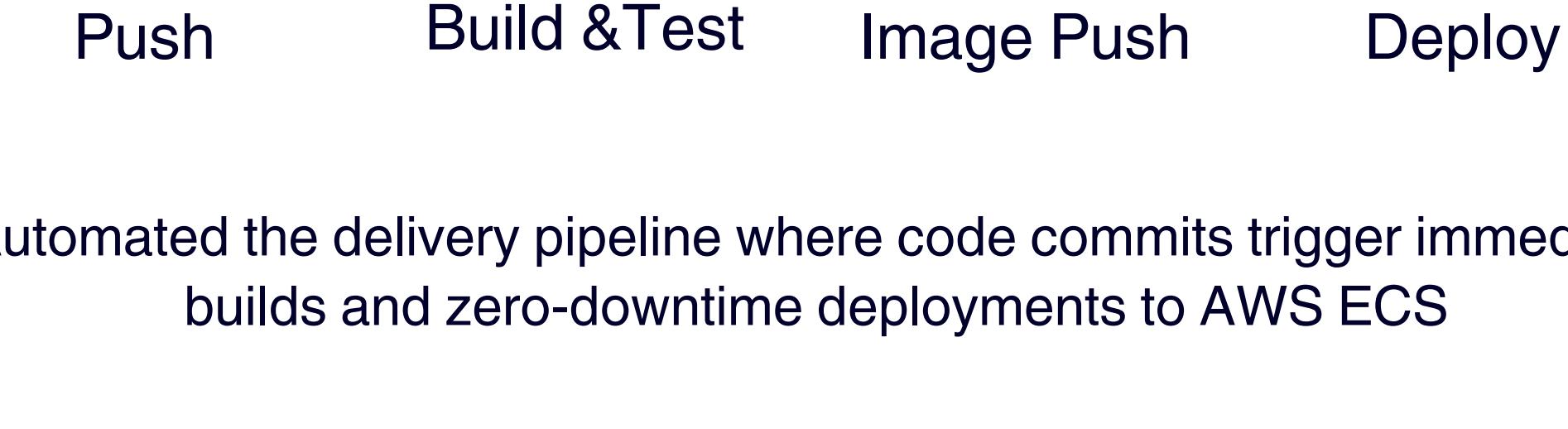
OBJECTIVES & ADVANTAGES

- Auto-scaling infrastructure using ECS Fargate
- Externalize media storage to Amazon S3.
- Isolate Database in private subnets.



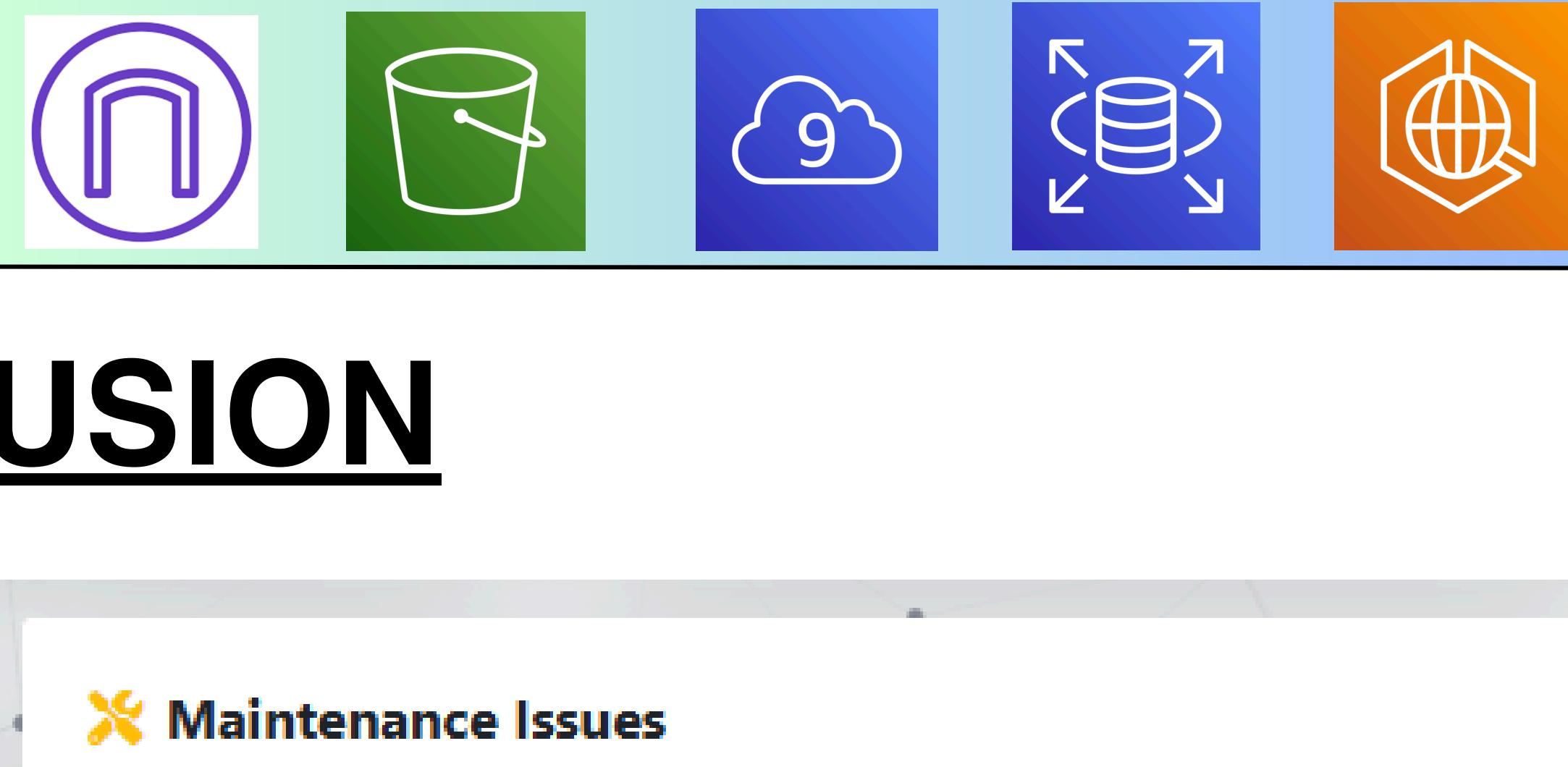
- Eliminates infrastructure management overhead using FARGATE.
- Deployed across Multi-Availability Zones (Multi-AZ).
- Database is inaccessible from the public internet (Private VPC).
- Implemented One-Time Password (OTP) logic via Google SMTP to prevent unauthorized account access.
- Bulk upload via excel file.

GITOPS WORKFLOW



Automated the delivery pipeline where code commits trigger immediate builds and zero-downtime deployments to AWS ECS

TOOLS & PLATFORM USED



CONCLUSION

Room Occupancy

Full (4)

Partial (1-3)

Empty (0)

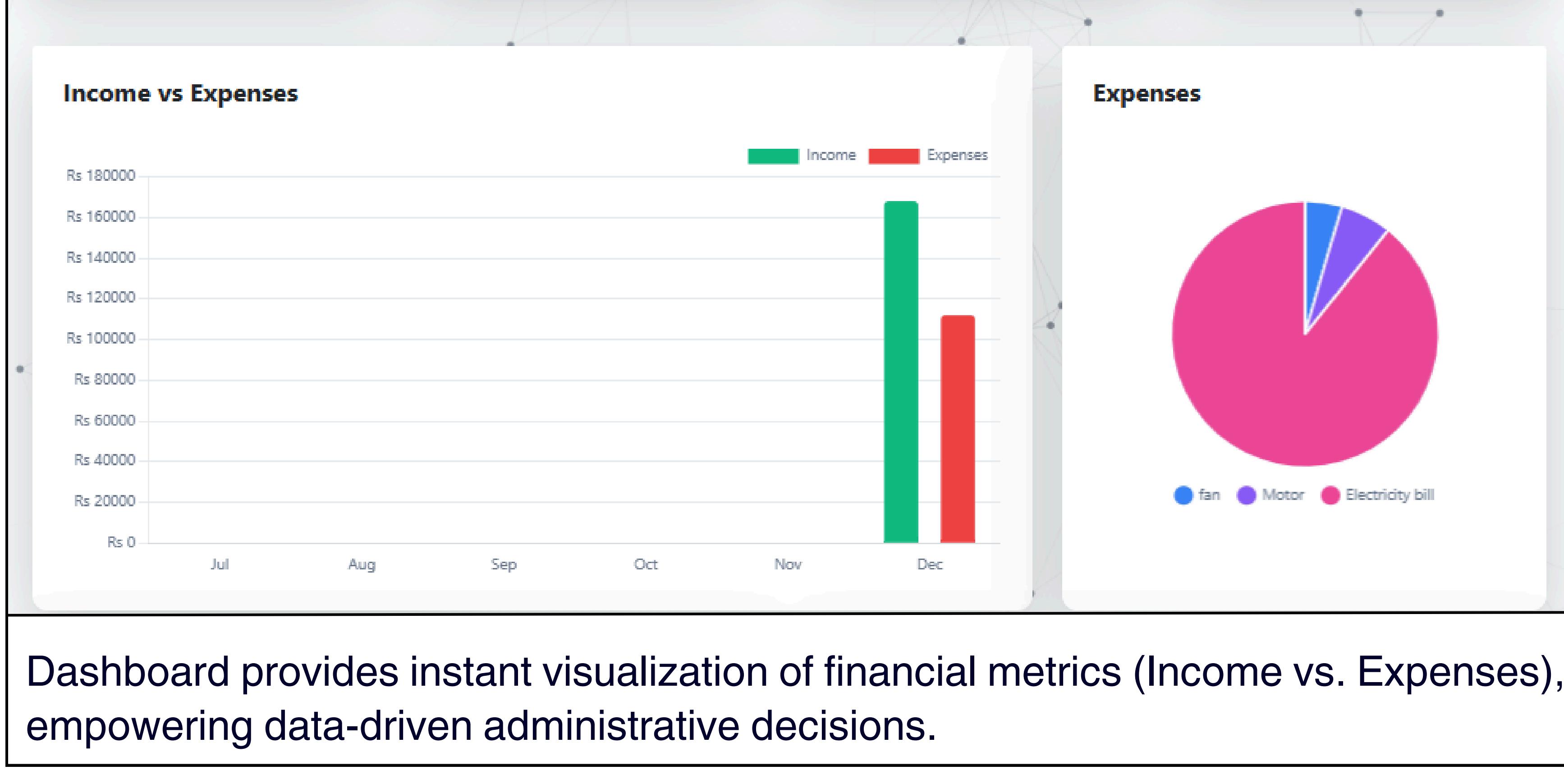
Maintenance Issues

Resolved

Pending

- Automated tracking of Room Occupancy and Maintenance Tickets significantly reduces manual record-keeping errors.
- Demonstrates seamless, low-latency data retrieval from the Private RDS Database to the Containerized Frontend.

This robust, cloud-native solution modernizes hostel administration by leveraging serverless AWS Fargate and automated GitOps pipelines to deliver a secure, scalable, and real-time management platform.



TEAM AND ACKNOWLEDGEMENT

Asma Haider
(BSSE23051)
Uzair Abdullah
(BSSE23075)

Dr. Zunnurain
Hussain
Sir Umair Makhdoom

REFERENCES

- Amazon Web Services (AWS) Documentation - ECS & Fargate.
- Docker Inc. - Containerization Best Practices.
- Flask Framework Documentation (Pallets Projects).
- GitOps Principles & GitHub Actions Guides.