Gwinnett Bus - Milestone 3

Team 3: Sam Bostian, Michael Rizig, Charlie Mclarty, Brian Pruitt, Allen Roman

Milestone 3

Containerization

Deployment

Documentation

Containerization

Podman

We chose Podman for containerization due to its security benefits and compatibility with Red Hat environments.

Application runs a Pod with a container for the producer-side and one for the consumer-side

Podman’s rootless containers provide an extra layer of security, which is ideal for production use.

Pod Configuration

The *pod.yml* file defines the pod structure, networking, and resource allocations, enabling smooth interaction between containers:

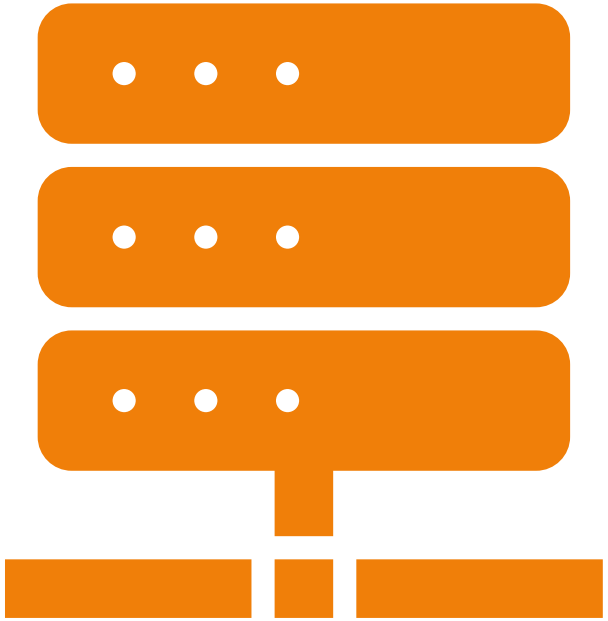
Producer Container (bus-project-producer)

- Handles Kafka event streaming, sending bus data to the topic.

Web Container (bus-project-web)

- Hosts the web interface for user interaction and data visualization.

Together, these containers operate within a Podman pod, ensuring efficient data flow and container management.



Deployment

Script Overview

The **run_pod.sh** script builds the required containers and deploys them using **podman play kube** based on the **pod.yml** configuration.

- This script ensures each container is started in the correct order with the right settings.
- If any part fails to build or deploy, the script immediately notifies us of the error.

Stopping and Cleaning up

The **stop_pod.sh** script stops and removes all running containers and pods, then performs a system prune to free up unused resources.

This cleanup process helps maintain an organized and efficient environment by removing any residual containers or resources.

Testing and Verification

Requirement	Description	Pass	Fail	Severity
Test Requirement	This is a Test Requirement to serve as an example	✓	✗	**Low**
Real-time data ingestion through Kafka	Verify that the Kafka producer sends data in real time	✓		**High**
Container initialization	Check if all containers start without errors	✓		**High**
Inter-container communication	Confirm Kafka-to-SQL data flow in pod environment	✓		**High**
Pod shutdown and cleanup	Ensure stop_pod.sh script removes all containers/pods	✓		**Medium**
Monitoring setup (optional feature)	Verify monitoring tool displays container status	✓		**Low**
Scalability test with 2,000 bus data points	Check system performance under high data volume	✓		**High**
Consistent container deployment	Test pod deployment on different environments	✓		**Medium**

Demo

Updated Data Simulation

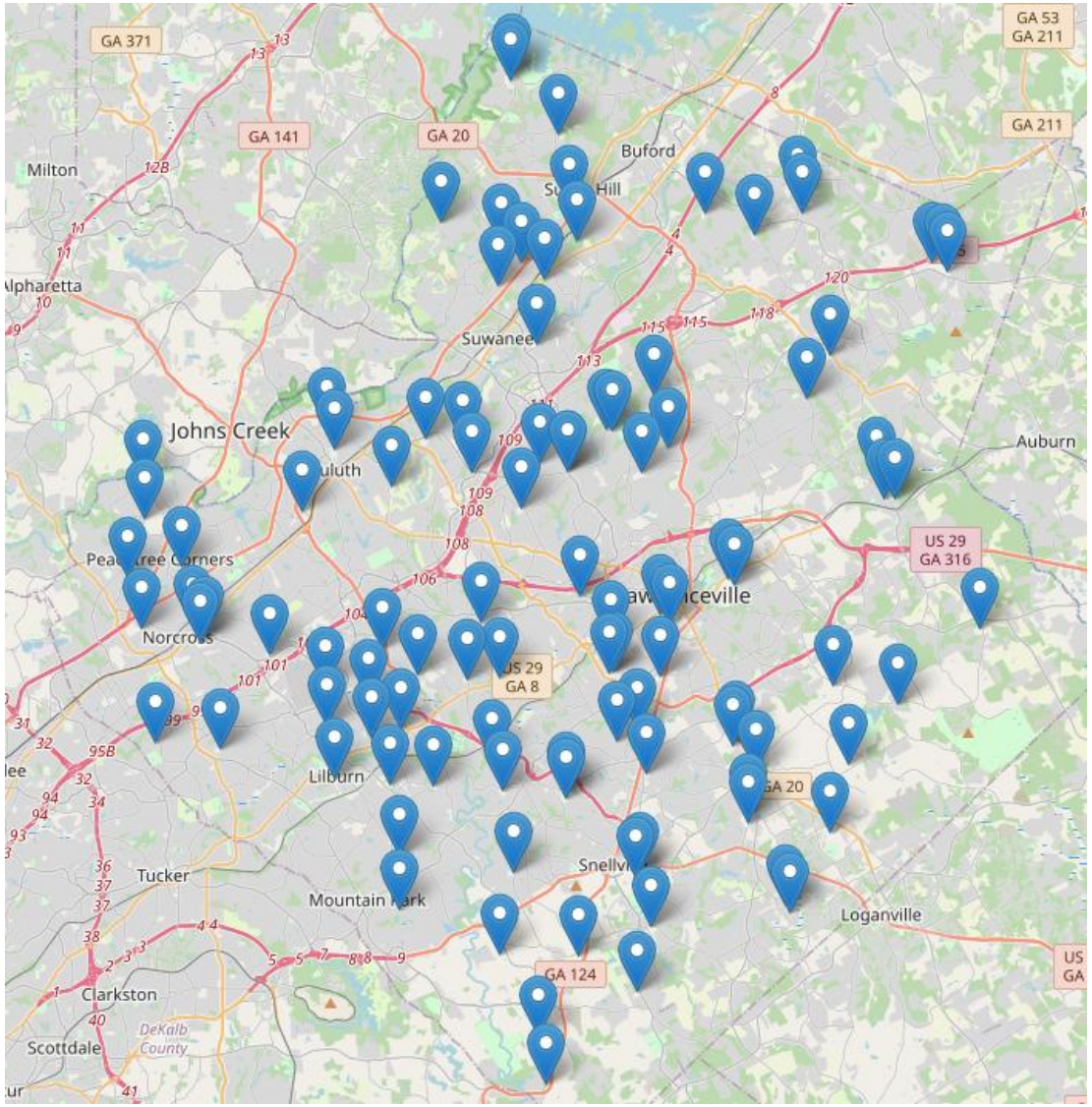
- All school coordinates saved and assigned to nearest node on graph

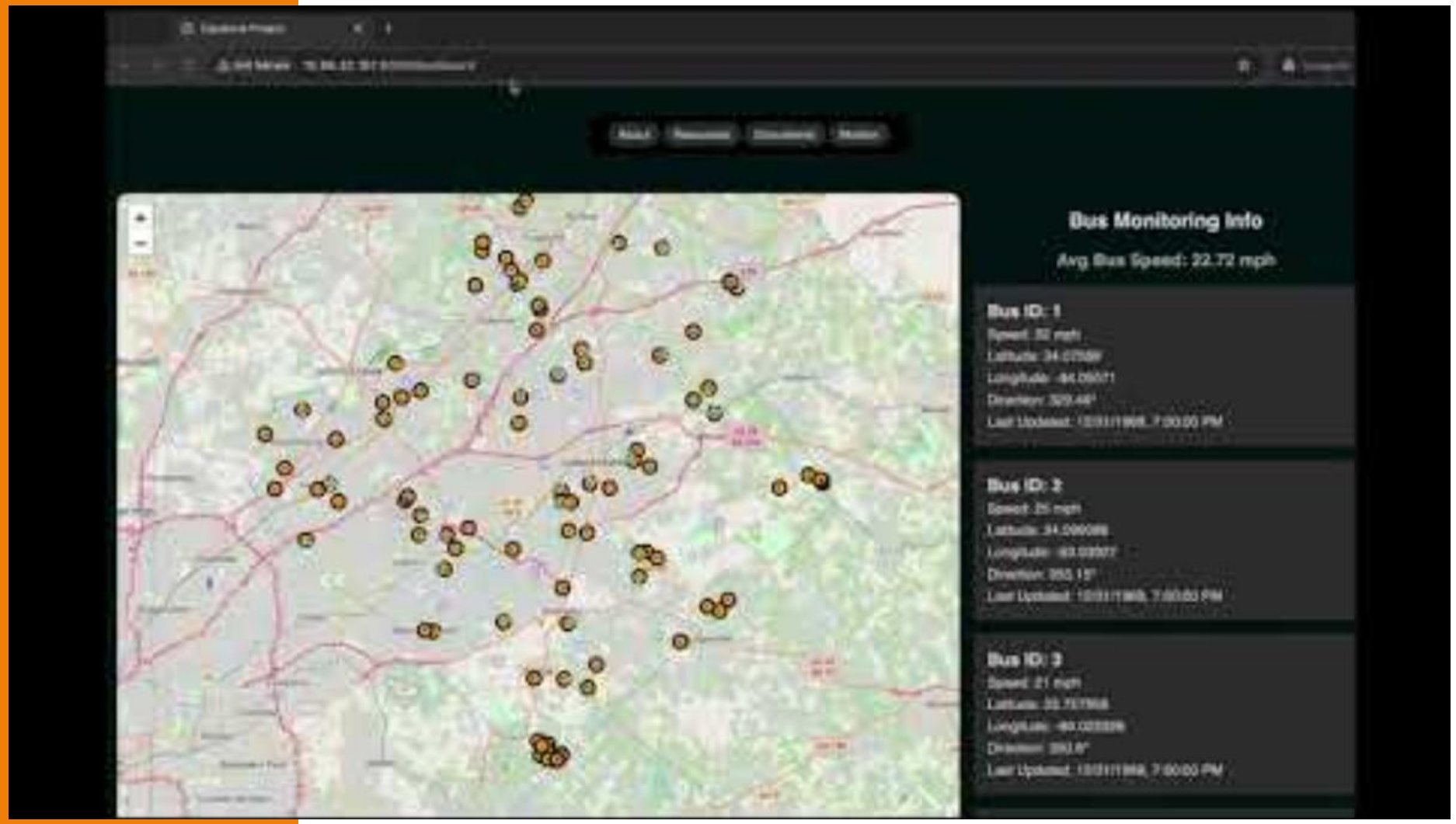
Routing:

- Each bus assigned random school to start their route
- A random walk occurs to create a route of ~3 miles

Timing:

- Simulation starts timer at 6:15 AM with launch of each bus




Bus Monitoring Info
Avg Bus Speed: 22.72 mph
Bus ID: 1
Bus ID: 2
Bus ID: 3

Documentation

Documentation

README.md Overview

- The README provides setup instructions, detailing installation, configuration, and initial deployment.

Configuration and Usage

- Instructions for modifying environment variables, scaling the system, and handling API requests are included to support GCPS's needs.

Optional Monitoring

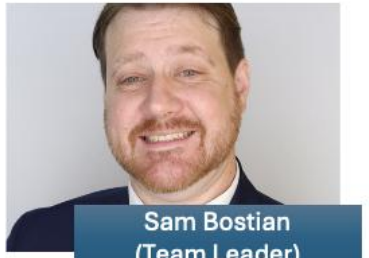
- Optional monitoring configuration is provided in the documentation, allowing GCPS to track container performance and ensure system stability.

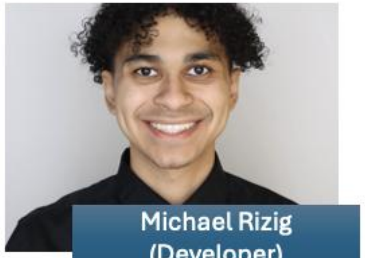
More Documentation!

- SDLC Documents included in the final deliverable, including Development Document, Software Test Plan/Report, Set-Up Documentation, and more!

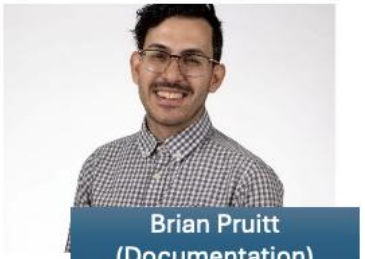
12-T3: GCPS REAL-TIME BUS MONITORING SYSTEM

SOFTWARE TEST PLAN & REPORT
CS 4850 - SECTION 01 – FALL 2024
NOVEMBER 10, 2024


Sam Bostian
(Team Leader)


Michael Rizig
(Developer)


Charlie Mclarty
(Developer)


Brian Pruitt
(Documentation)


Allen Roman
(Developer)

Project Recap

Key Achievements

Consistent Deployment

Podman containers and pod configurations ensure a stable deployment process across environments.

Efficient Management

Automated setup and teardown scripts simplify container management.

Comprehensive Documentation

Detailed documentation supports future maintenance and scalability.


Questions?