Spotter

Team 4: Sean Whoriskey, David Rockwell, Zachary Smith, Mike Collins, and Brianna Ayala

https://github.com/seanwho88/CSC468Project4.git

Chapter 1

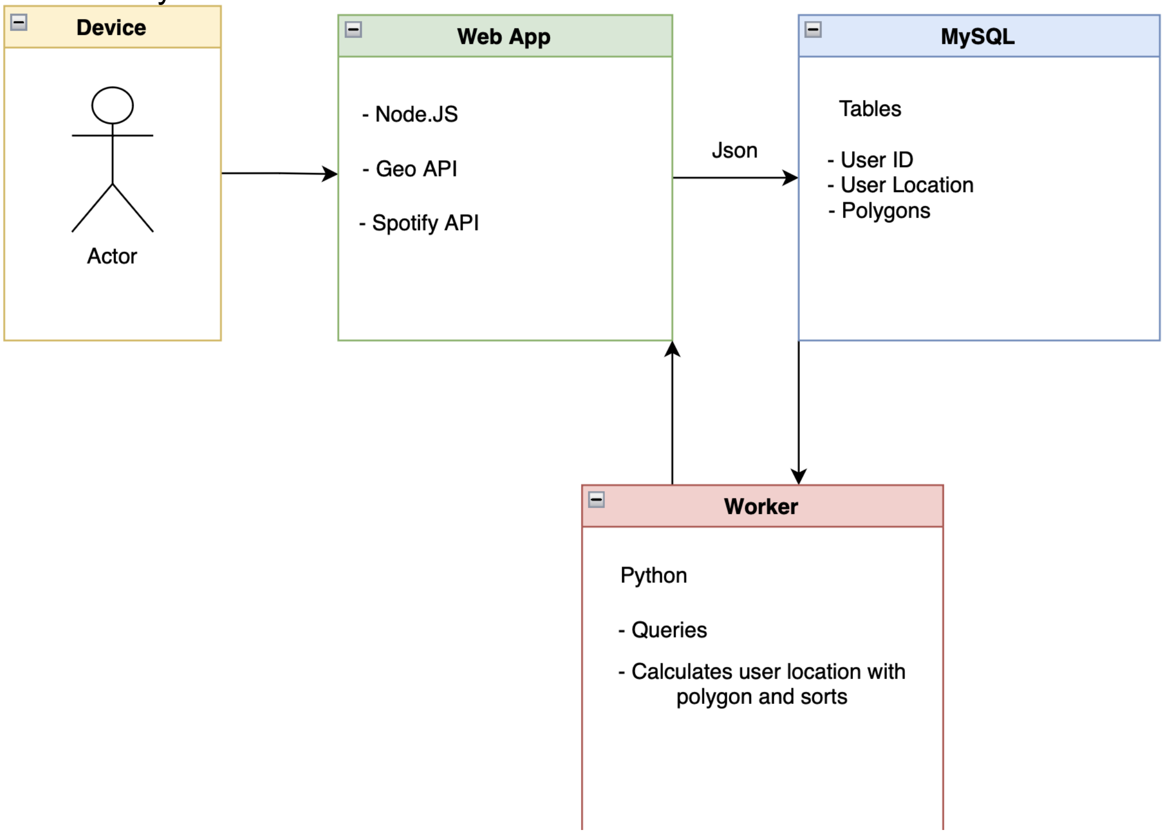
Team Vision/Design

Introduction: The goal of this project is to develop and deploy a cloud-based music sharing web application that allows users to connect with others who share their musical interests. The app will utilize real-time geolocation to find other users in proximity and display the music they are currently listening to.

Features:

- **Connect with Spotify:** Users will connect their Spotify account to the app, which will then display their current playing song for other users to see.
- **Geolocation-based Sharing:** The app will use real-time geolocation to find other users in proximity and display their currently playing song.
- Social Interaction: As a long term goal, users will be able to like, comment, or follow other users, providing a way to socialize through music.
- **Music Discovery:** The app will provide a new way for users to discover new music by browsing other users' currently playing songs.
- **User Interface:** The app will have a simple, intuitive interface, allowing users to quickly connect their Spotify account and start sharing their music.

Conclusion: Spotter (name in progress) is a fun and innovative way to connect with others who share your musical interests. With its real-time geolocation and social interaction features, it provides a new way to discover new music and socialize through the music you love.



Chapter 2

Team Design Implementation

JavaScript and Node.js: JavaScript will be used as the primary programming language for building the front-end user interface and the backend of the app. Node.js will be used as the runtime environment to execute JavaScript on the server-side. This will allow the app to handle multiple concurrent user requests.

Geolocation API: The app will use a Geolocation API to retrieve the real-time location of the user. This information will then be stored in the database so that calculation of user proximity can be calculated by the worker.

Spotify API: The app will use the Spotify Web API to retrieve the current playing song of the user, which will be stored and updated in the database.

MySql: MySql will be used as the database for storing user data, including their user location and current playing song from Spotify.

Python: Python will be used to implement the backend logic for handling user requests and interacting with the Spotify and Geolocation APIs. Python will also be used for any data processing tasks, such as filtering and aggregating data from the MySQL database.

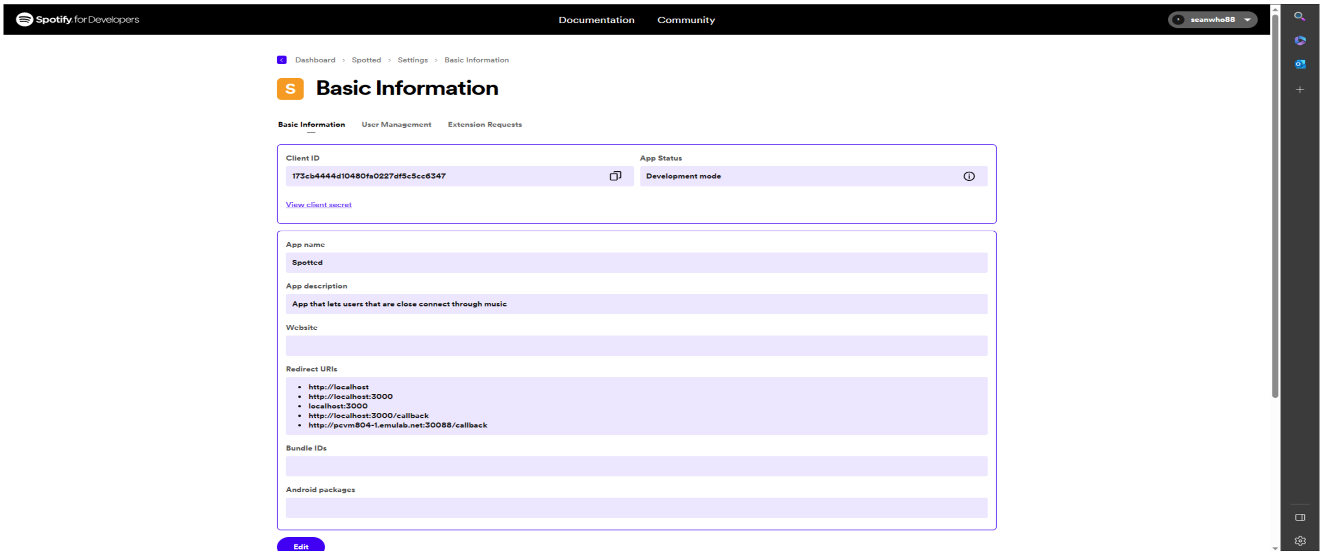
Overall, the implementation of the web application, Spotter (in progress) will use a combination of JavaScript and Node.js for the front-end and back-end, a Geolocation API for retrieving real time location and storing on the database.

To begin we will start small-scale with manually inputting data and test with one set location. We are able to expand into more extensive areas and data as we meet the small goals locally.

Current State of Spotter

4/15/2023

One of the main issues with our project was its ambitious use of API's to complete our desired goal. The geolocation API and Spotify API were huge bottlenecks for making any progress. We had to look through both API's extensively to be able to use them to any degree of success. Even after that, the Spotify has a 25 user limit for each app you create as well as additional blocks like the required redirect link:



Client ID
173cb4444d10480fa0227df5c5cc6347

App Status
Development mode

App name
Spotted

App description
App that lets users that are close connect through music

Website

Redirect URIs
- http://localhost
- http://localhost:3000
- localhost:3000
- http://localhost:3000/callback
- http://pcvm804-1.emulab.net:30088/callback

Bundle IDs

Android packages

In order to authorize correctly we need a redirect link that is valid, but this changes every time we launch a new experiment, thus we can't necessarily automate this process. Instead, we have to manually enter our redirect URL into our Spotify's developer dashboard. In addition, the geolocation API requires a secure website to access location on a non-local hosted site. Once again, the hostname changes every time we make a new cloud project and we were concerned about setting up automation for an SSL certificate within the time frame of the project. To bypass this we are currently using google chrome beta tools to white-list the website in order to access the geolocation. While we have worked around our issues with the Spotify and the Web Geolocation API, we are still concerned with achieving the functionality we desire by the end of the semester.

Moving on from the front-end (node.js) issues, the python worker in the background has only a few issues left to iron out. We have a good proof of concept where we use Python and sql packages in order to manipulate the database. We are planning on using location IDs for each building and checking if the user is within a certain radius of the location IDs center longitude and latitude. This is not fully working, because it is difficult to cleanly work with the data within the full-stack. We are very close to having this completely usable though.

Project Feasibility

With regards to the “A-level technical requirements” from the Course Project section on d2l, we firmly believe that Spotter is well on its way to having all of the functionality for Docker, K8, and Jenkins set up by the end of this semester. At some point we would like to add a branch for kubernetes and helm so that we do not have to rely on instantiating our project from Dr. Ngos CSC468 profile on CloudLab.

With regards to the actual functionality of the app we are pretty worried about its completion by the end of the semester. While we have some aspects of the app working we do not have the logic of the app meshing together quite yet.

Chapter 3

Docker Images

Web App: We created the Dockerfile for the “webapp” component using the node:14-alpine image. The Dockerfile sets up the work directory to /webapp and copies any package.json files. The necessary dependencies for the web app are then installed. The application files are then copied into the container, and the application is exposed on port 3000. We use CMD to run npm start at the end.

We ran into issues regarding our connection to the database through the web app as the database’s container took longer to initialize than the web app. To solve this we added a “depends-on” statement to the “webapp” service in our docker-compose.yaml file. We then added a loop to our web app to continuously attempt connection to the database until everything is initialized. Overall, the procedure for setting up this Dockerfile was relatively simple.

Database: We created the Dockerfile for the “database” component using the official mysql/mysql-server image, which provides our MySQL environment. We set environment variables for the root password, database name, database user, and user password. This information will be put into environment variables in the future for privacy reasons. Finally, we made a custom SQL script (init.sql) that is copied into the /docker-entrypoint-initdb.d/ directory. This script is executed during the container startup.

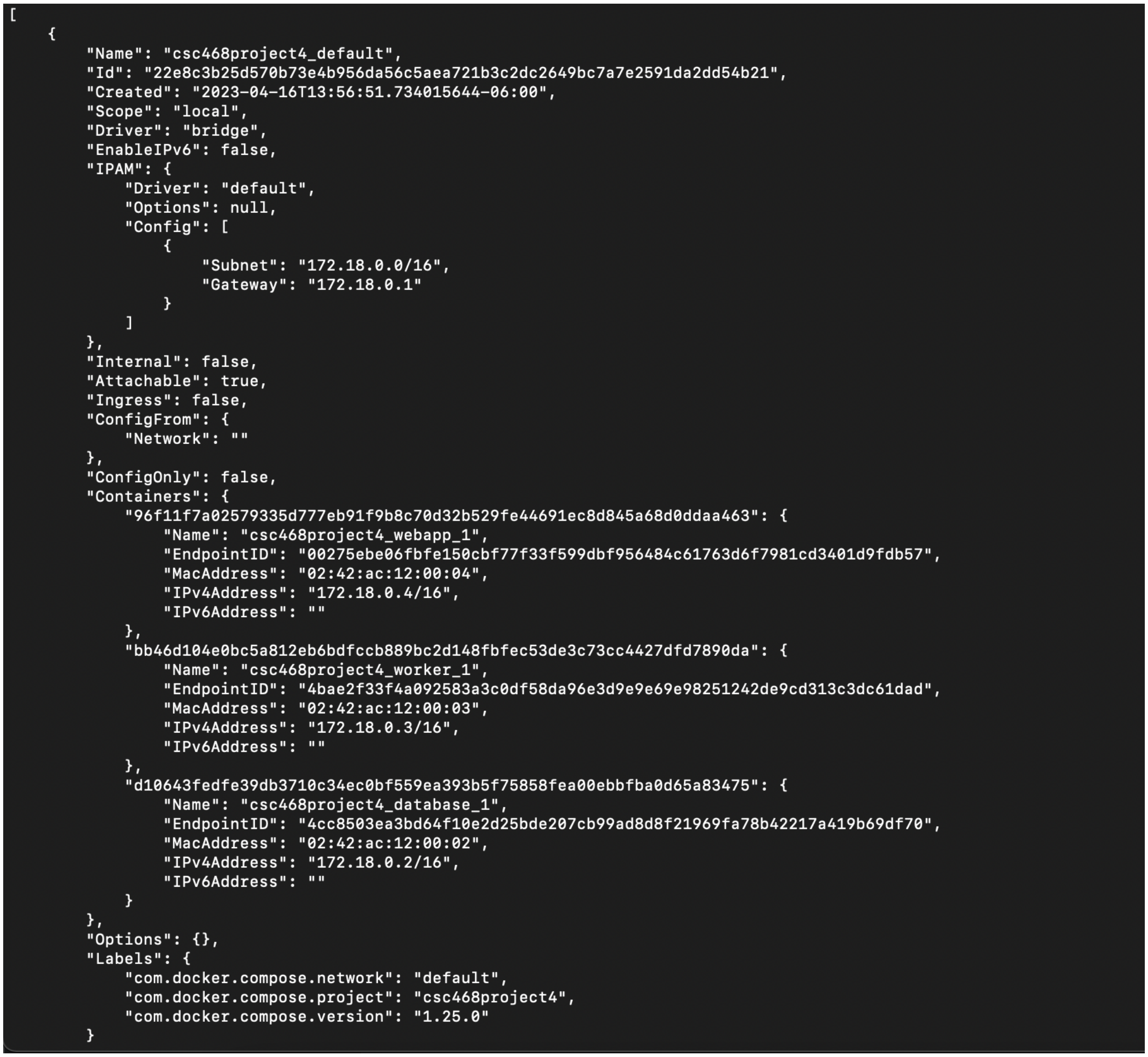
Overall, the procedure for building this Dockerfile was simple as well. As we worked on designing our database, the init.sql script continuously changed. At this point in the project, the init.sql script builds our users and location tables.

Worker: We created the Dockerfile for our “worker” component using the python:3.9 official image. The working directory is set to /worker, and the requirements.txt file is copied into the container. The required Python packages are installed through that requirements.txt file, and the application files are copied into the container. We expose the worker on port 5000, and then use CMD to start the worker using the run.sh script.

The creation of the “worker” Dockerfile has been the most challenging so far, as we have faced troubles with orchestrating the multiple files that make up our worker component. For example, as new files were added we created a script “run.sh” to start all of the worker scripts when the container is built.

Chapter 4:
Docker Networking

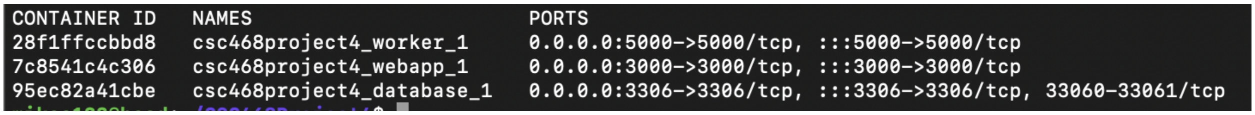
Our docker-compose.yaml file defines our service's exposed ports. We map port 3000 on the host machine to port 3000 in our "webapp" container. We do the same for port 3306 and 5000 for our database and worker respectively. By default, Docker Compose creates a bridge network for the services defined in the compose file. In our case, you can see the network is named "csc468project4_default". Internally, the "webapp" and "worker" services can communicate with the database using the unique ip address for each container and the exposed port 3306. In the image below you can see more details regarding our project's docker network:



```
[
    {
        "Name": "csc468project4_default",
        "Id": "22e8c3b25d570b73e4b956da56c5aea721b3c2dc2649bc7a7e2591da2dd54b21",
        "Created": "2023-04-16T13:56:51.734015644-06:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.18.0.0/16",
                    "Gateway": "172.18.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": true,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "96f11f7a02579335d777eb91f9b8c70d32b529fe44691ec8d845a68d0ddaa463": {
                "Name": "csc468project4_webapp_1",
                "EndpointID": "00275ebe06fbfe150cbf77f33f599dbf956484c61763d6f7981cd3401d9fdb57",
                "MacAddress": "02:42:ac:12:00:04",
                "IPv4Address": "172.18.0.4/16",
                "IPv6Address": ""
            },
            "bb46d104e0bc5a812eb6bdfccb889bc2d148fbfec53de3c73cc4427dfd7890da": {
                "Name": "csc468project4_worker_1",
                "EndpointID": "4bae2f33f4a092583a3c0df58da96e3d9e9e69e98251242de9cd313c3dc61dad",
                "MacAddress": "02:42:ac:12:00:03",
                "IPv4Address": "172.18.0.3/16",
                "IPv6Address": ""
            },
            "d10643fedfe39db3710c34ec0bf559ea393b5f75858fea00ebbfba0d65a83475": {
                "Name": "csc468project4_database_1",
                "EndpointID": "4cc8503ea3bd64f10e2d25bde207cb99ad8d8f21969fa78b42217a419b69df70",
                "MacAddress": "02:42:ac:12:00:02",
                "IPv4Address": "172.18.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {
            "com.docker.compose.network": "default",
            "com.docker.compose.project": "csc468project4",
            "com.docker.compose.version": "1.25.0"
        }
```

Our attempts to map our components went smoothly for the most part. We had issues with orchestrating the database container's initialization and the webapp's connection to the database. We fixed this by continuously attempting to connect to the database until a connection is finally made. We also ran into issues setting up the hostname for our web app. We fixed this by creating a script called "hostname.sh". This script is used to change any instance of the word "hostname" or "mysql" to the actual hostname of our machine. This streamlined our ability to instantiate our project on Docker.

Docker Container Port Mapping:

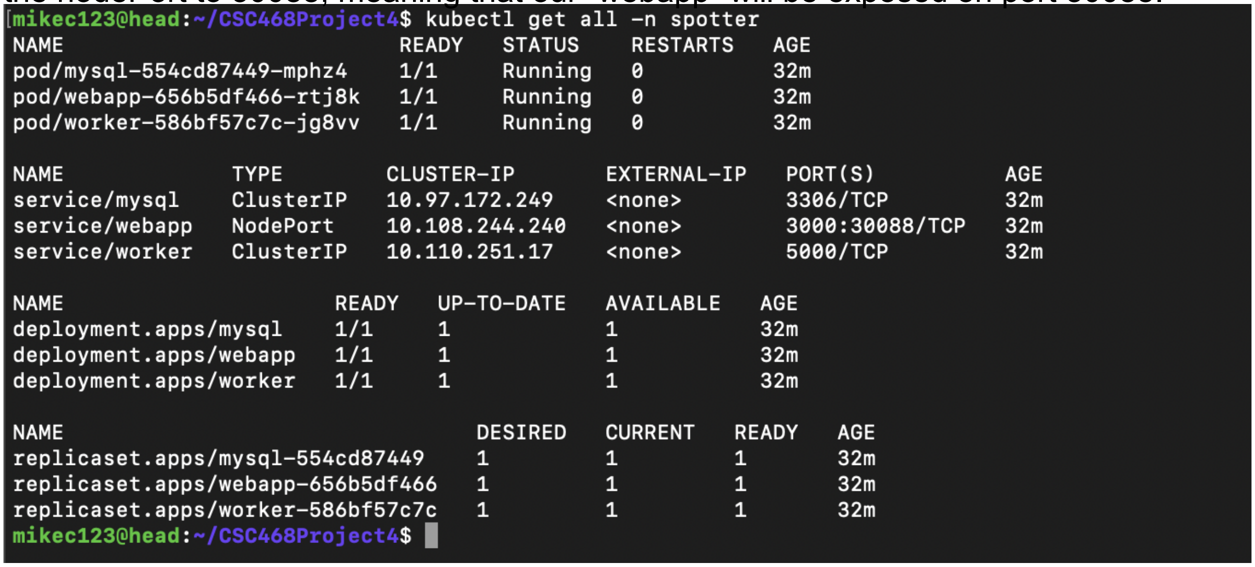


```
CONTAINER ID   NAMES                         PORTS
28f1ffccbbd8   csc468project4_worker_1       0.0.0.0:5000->5000/tcp, :::5000->5000/tcp
7c8541c4c306   csc468project4_webapp_1       0.0.0.0:3000->3000/tcp, :::3000->3000/tcp
95ec82a41cbe   csc468project4_database_1     0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060-33061/tcp
```

Chapter 4.1:

Kubernetes Networking

Currently kubernetes pulls our images from our project's docker hub repository and sets up our containers with the same port mapping as mentioned earlier. In our kubernetes service file we set up our mysql and worker services using a ClusterIP - meaning the services "mysql" and "worker" will only be accessible within the kubernetes cluster. Our "webapp" service uses NodePort for its networking capabilities. The service listens on port 3000 and its targetPort is also set to 3000. The service.yaml file also sets the nodePort to 30088, meaning that our "webapp" will be exposed on port 30088.



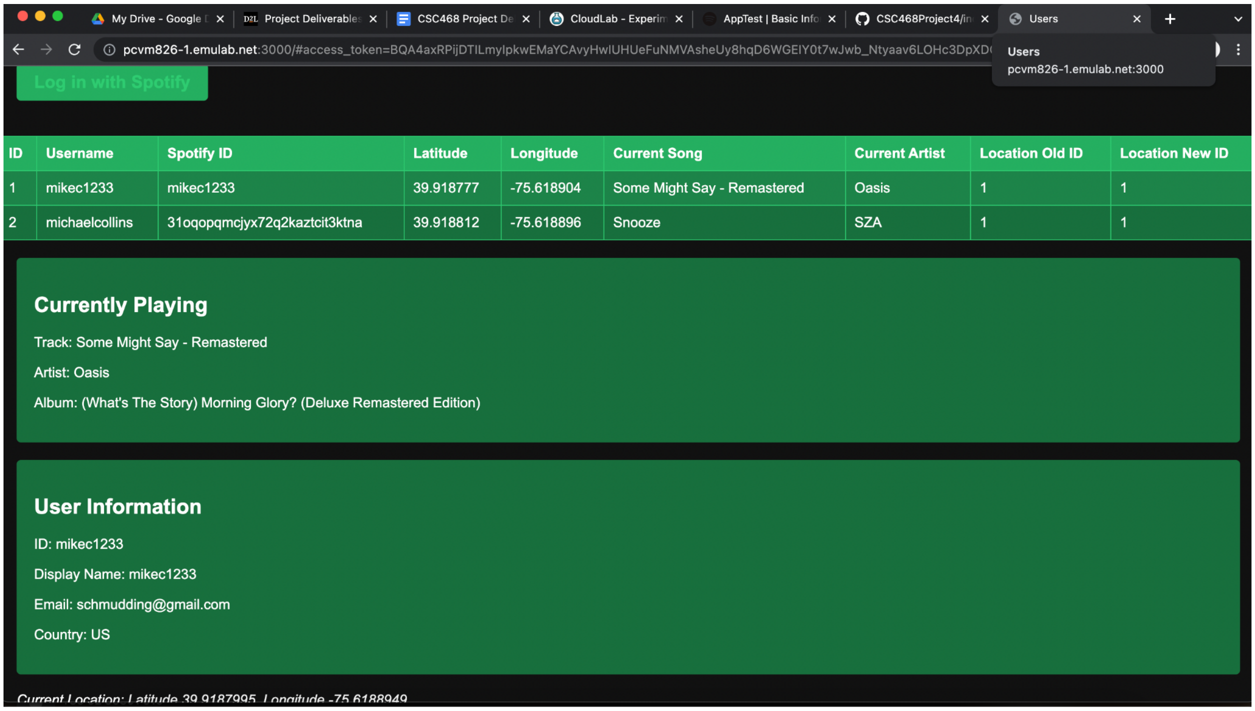
```
[mikec123@head:~/CSC468Project4$ kubectl get all -n spotter
NAME                             READY   STATUS    RESTARTS   AGE
pod/mysql-554cd87449-mphz4       1/1     Running   0          32m
pod/webapp-656b5df466-rtj8k      1/1     Running   0          32m
pod/worker-586bf57c7c-jg8vv      1/1     Running   0          32m

NAME               TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
service/mysql      ClusterIP   10.97.172.249    <none>        3306/TCP         32m
service/webapp     NodePort    10.108.244.240   <none>        3000:30088/TCP   32m
service/worker     ClusterIP   10.110.251.17    <none>        5000/TCP         32m

NAME                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/mysql    1/1     1            1           32m
deployment.apps/webapp   1/1     1            1           32m
deployment.apps/worker   1/1     1            1           32m

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/mysql-554cd87449    1         1         1       32m
replicaset.apps/webapp-656b5df466   1         1         1       32m
replicaset.apps/worker-586bf57c7c   1         1         1       32m
mikec123@head:~/CSC468Project4$
```

Current State of the app with two unique users.


My Drive - Google D
Project Deliverables
CSC468 Project De
CloudLab - Experim
AppTest | Basic Info
CSC468Project4/in
Users
pcvm826-1.emulab.net:3000/#access_token=BQA4axRPijDTILmylpkwEMaYCAvyHwIUHUeFuNMVAsheUy8hqD6WGEIY0t7wJwb_Ntyaav6LOHc3DpXD(
Users
pcvm826-1.emulab.net:3000
Log in with Spotify

ID	Username	Spotify ID	Latitude	Longitude	Current Song	Current Artist	Location Old ID	Location New ID
1	mikec1233	mikec1233	39.918777	-75.618904	Some Might Say - Remastered	Oasis	1	1
2	michaelcollins	31oqopqmcjyx72q2kaztcit3ktna	39.918812	-75.618896	Snooze	SZA	1	1

Currently Playing

Track: Some Might Say - Remastered

Artist: Oasis

Album: (What's The Story) Morning Glory? (Deluxe Remastered Edition)

User Information

ID: mikec1233

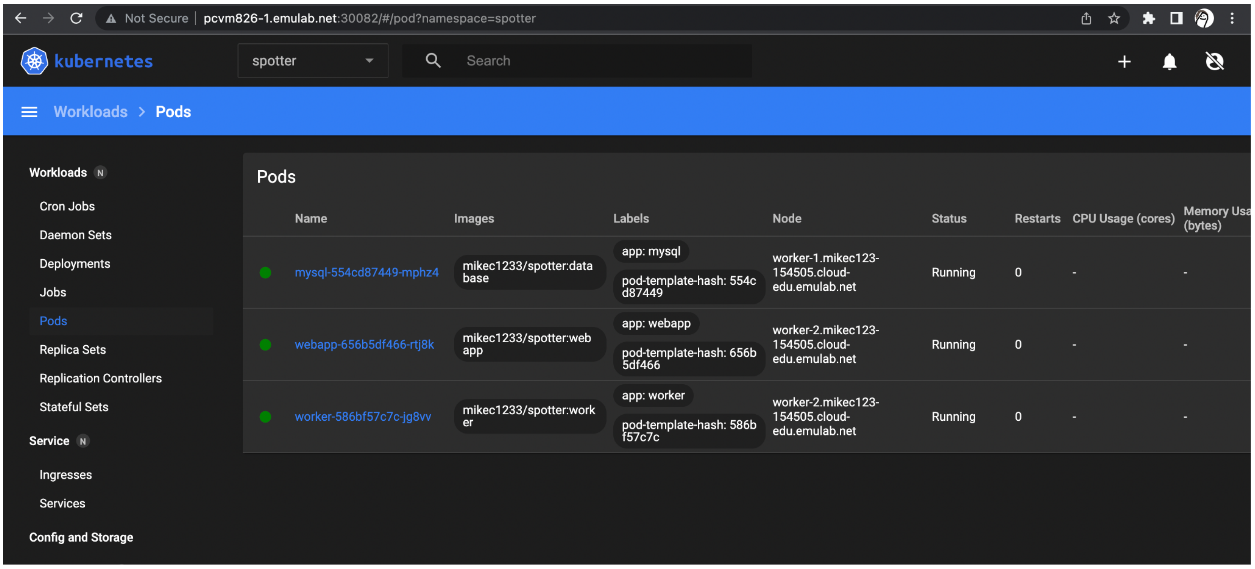
Display Name: mikec1233

Email: schmudding@gmail.com

Country: US

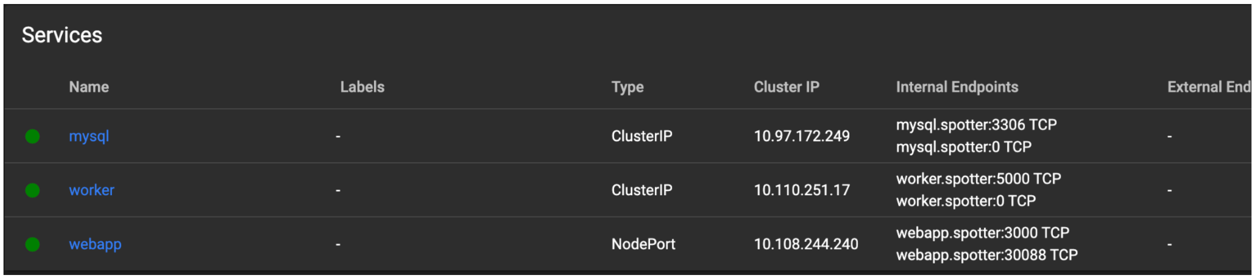
Current Location: Latitude 39.9187995, Longitude -75.6188949

Kubernetes Dashboard: Pods and Services Information.



Pods

	Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usa (bytes)
●	mysql-554cd87449-mphz4	mikec1233/spotter:database	app: mysql pod-template-hash: 554cd87449	worker-1.mikec123-154505.cloud-edu.emulab.net	Running	0	-	-
●	webapp-656b5df466-rtj8k	mikec1233/spotter:webapp	app: webapp pod-template-hash: 656b5df466	worker-2.mikec123-154505.cloud-edu.emulab.net	Running	0	-	-
●	worker-586bf57c7c-jg8vv	mikec1233/spotter:worker	app: worker pod-template-hash: 586bf57c7c	worker-2.mikec123-154505.cloud-edu.emulab.net	Running	0	-	-



Services

	Name	Labels	Type	Cluster IP	Internal Endpoints	External End
●	mysql	-	ClusterIP	10.97.172.249	mysql.spotter:3306 TCP mysql.spotter:0 TCP	-
●	worker	-	ClusterIP	10.110.251.17	worker.spotter:5000 TCP worker.spotter:0 TCP	-
●	webapp	-	NodePort	10.108.244.240	webapp.spotter:3000 TCP webapp.spotter:30088 TCP	-

David Rockwell

270 Valley Road, Media, PA 19063

davidrockwell7310@gmail.com | 484-844-4674

linkedin.com/in/david-rockwell-83b49222b

EDUCATION

West Chester University | West Chester, Pennsylvania | August 2021 - May 2023 (expected)

- Bachelor of Science, Computer Science, Specialization: Cyber Security
- GPA: 3.5

Delaware County Community College | Media, Pennsylvania | August 2019 – May 2021

WORK & INTERNSHIP EXPERIENCE

TA Technologies | New Castle, Delaware | May 2022 – August 2022

Software Developer Intern

- Developed an automated Azure DevOps CD/CI pipeline for code generation and test tool automation.
- Created Azure Cloud virtual machines (VM) to test before deployment.
- Application deployment and automated testing using Test Complete as part of CD/CI pipeline.
- Made heavy use of PowerShell to implement Azure DevOps environment.
- Implemented SSH based password-less communication between pipeline and VMs.

DR Technologies | Media, Pennsylvania | September 2021 – January 2022

Software Test Engineering Intern

- Developed an IoT mesh network to monitor temperature and events for warehouse application.
- Worked on Python backend and Javascript frontend of system.
- Tested device configuration and calibration interfaces and helped debug the integrated system.
- Used virtual machines to run server code and debug Python application.
- Used developer tools through browser to debug user interface.

West Chester University | West Chester, Pennsylvania| January 2022 - Present

IT Help Desk Consultant

VOLUNTEER & INVOLVEMENT

Computer Science Club | West Chester, Pennsylvania | September 2021 – Present

- Work with other club members on computer science related projects
- Discuss different topics in the Computer Science field

Cyber Security Club | West Chester, Pennsylvania | September 2021 – Present

- Participate in local CTF events
- Discuss news stories and emerging topics in the Cyber Security field

First Robotics Competition | Media, Pennsylvania | September 2013 – April 2022

- Participated in robotics competitions from Middle School through High School
- During college, volunteered mentoring younger competitors

TECHNICAL SKILLS

Languages and Applications: Java, C++, C, Python, Javascript, HTML, VirtualBox, SceneBuilder, NetBeans, IntelliJ, Visual Studio Code, JSON serializer/deserializer, Flask server, Bootstrap, web sockets, PowerShell, Azure DevOps, Azure Cloud, Bomgar, ServiceNow, Virtual Machines, SSH

ZS

Contact Information:

- 8 Chisel Creek Drive
  Landenberg, PA 19350
- Zsmi57@gmail.com
- 610-800-4473

Relevant Skills:

- Organized
- Collaborative Worker
- Detail Oriented
- Problem Solver
- Flexible
- Independent
- Multitasker
- Easy to work with
- Passionate

Certifications/Clearances:

- Projected Cyber Security Certificate
- Projected CompTIA Security + Cert
- Microsoft Applications
- DOD Government Secret Clearance (2023-Current)

Projected Degree:

- Honors Seminar Program
- Major Computer Science
- Minor Criminal Justice

LinkedIn

Zachary J. Smith

Summary:

Currently a Junior at West Chester University striving for a major in computer science with a concentration in cyber security and a minor in criminal justice. I am an active board member and co-founder of the West Chester University Cyber Security club.

Recently I have had the opportunity to work as an intern for Children's Hospital of Pennsylvania (CHOP) as an information security intern working in a large hospital environment. This allowed me to apply my current studies to real life applications and further advanced my understanding in this field. I learned how to manage many different software programs and applications to give the proper access to CHOP employees and staff. In addition, I maintained the device workstations inside the CHOP locations to ensure that they were properly managed and secure.

This experience has been very beneficial in my goal to continue my pursuit in the cyber security industry. I continue to seek more hands-on training and knowledge to build off my education and future studies in cyber security.

Education:

West Chester University *West Chester, PA* (August' 20 – Current)
****Anticipated Graduation 'December' 23***

- Cum GPA 3.847 / Dean's List (Every Semester) / Honors Program
- Co-Founder/Secretary Cyber Security Club

Avon Grove High School * (West Grove, PA * June' 20)

- High Honor-Roll/GPA 3.89
- President and co-founder of FBLA (Future Business Leaders of America

Work Experience:

Children's Hospital of Philadelphia * (June'22-Current)

- Information Security Intern (Security Account Admin Team)

US Soccer Certified Soccer Official/Assignor * (March'17 – Current)

- Referee of The Year Award EPYSA/State of Pennsylvania * 2019
- NCAA Official * (January'19 – Current)
- Regional Referee *(January'21-Current)
- Nationals Championship Referee *(July '22)

PIAA Certified Wrestling Referee *(January'21 – Current)

Extracurricular Experience:

West Chester University Cyber Security Club * (January'20-Current)

West Chester University Computer Science Club * (January'20-Current)

Avon Grove High School Wrestling Team * (November'16 – March'20)

- Varsity Wrestler 126lbs-155lbs/Captain

Hack-The-Box Modules

NSA Codebreaker Challenges

14 Shannon Dr,
Chester Springs, PA 19425
(484) 905 - 2035
ayalanbrianna@gmail.com

Brianna Ayala

EDUCATION

West Chester University — *Bachelor of Science in Computer Science*

- August 2021 – Expected May 2023, West Chester

Pennsylvania State University — *Bachelor of Science in Computer Science*

- August 2017 – May 2021, Penn State Brandywine & Harrisburg
- Jane E. Cooper Honors Program
- Engineering Club: Worked on designing parts for a drone and telescope using SolidWorks as a team to 3d-print and assemble the pieces of these designs

SKILLS

- Operating Systems: Windows, MacOS & Linux
- Programming languages: C++, Java, Python, C, R, SQL, HTML & Matlab
- Scripting: PowerShell
- Cloud Technologies: Oracle, VMWare & Microsoft Azure
- Security tools: InsightVM, Rapid7, KnowBe4 & Brinqa
- Penetration testing: Kali Linux
- Design: SolidWorks & Cura
- Frameworks: Spring Framework

EXPERIENCE

Qurate Retail Group, West Chester — *Cybersecurity - PCI DSS & Vulnerability Management Intern*
June 2022 – Present

- Assist with Payment Card Industry (PCI) audits
- Review Payment Card Industry (PCI) Data Security Standard Report on Compliance
- Work with the vulnerability management team to perform daily fire drill assessments for new critical vulnerabilities and patches
- Scan assets for vulnerabilities using Rapid7 & InsightVM

Spring Education Group, West Chester — *IT & Cybersecurity Intern*
January 2022 – June 2022

- Assisted with writing security policies
  Tested & implemented company-wide use of required multi-factor authentication
- Monitored the MFA help desk email
- Wrote python script to perform automated user reconciliation processes to identify inactive employee accounts

Target, Phoenixville — *Guest Service Advocate*
August 2016 – January 2022

- Provided quality customer service efficiently to ensure customer satisfaction
- Worked to solve any issues that customers may have regarding items purchased, inventory, pricing, policies, etc.
- Handling phone calls and directing to the appropriate department necessary as well as making storewide announcements when necessary

Sean Whoriskey

610-554-2753 | seanwho88@gmail.com | linkedin.com/in/seanwhoriskey

Education

West Chester University West Chester, PA
Bachelor of Arts in Computer Science, Minor in Computational Science *Aug. 2020 – May 2023*

Experience

Software Developer Intern May 2022 – August 2022
Savana Inc. *Malvern, PA*

- Worked on DNA banking core integration project utilizing OutSystems and Postman
- Completed tickets in a scrum-based work environment
- Worked on platform engineering and banking services API team

Computer Science Tutor Jan. 2021 – Present
West Chester University *West Chester, PA*

- Tutored computer science II and III
- Tutored computer systems
- CRLA certified tutor level II

Store Team Member May 2019 – July 2019
Sheetz inc. *Morgantown, PA*

- Managed register while keeping customers entertained and moving swiftly
- Helped manage kitchen duties and general store workflow
- Increased customer satisfaction score

Projects

Yelp Review Data Analysis | *Python, Pyspark, Jupyter Notebook, Markdown* Sep 2022 – December 2022

- Used Jupyter Notebook and Kaggle to conduct data analysis on Yelp reviews
- Analyzed millions of users, reviews, and businesses
- Visualized users eating patterns and locations using a plotly heat map
- Used multiple skills to sift through JSON data using RDD's

Class Hierarchy Analyzer | *Java, Eclipse* Sep 2022 – December 2022

- Used Java meta-programming
- Created a program that takes in a Java project and returns the class hierarchy
- Utilized tree searching algorithms and data structures

Activities

Esports Club Rocket League Game Lead Sep 2022 – Present

- Organized and participated in tournaments each semester (NACE and RLCS)
- Scheduled practices and scrimmages against other college teams

West Chester Competitive Programming Competition October 2022

- Participated in Java programming competition
- Completed all three problems within 45 minutes and won $15 in Amazon gift cards

West Chester Computer Science Meet and Greet September 2022

- Participated in Java programming competition
- Completed all three problems within 45 minutes and won $15 in Amazon gift cards

West Chester Computer Science Meet and Greet September 2022

- Participated in Java programming competition
- Completed all three problems within 45 minutes and won $15 in Amazon gift cards

Technical Skills

Languages: Java, Python, C/C++, MATLAB
Developer Tools: Git, VS Code, Visual Studio, PyCharm, IntelliJ, Eclipse
Libraries: pandas, NumPy, Matplotlib

Michael Collins

LinkedIn | 336-769-7741 | mcolls1233@gmail.com

Skills

- JAVA | C | HTML/CSS | JavaScript | Node.js | Express.js | MongoDB | SQL | Docker | Kubernetes | Git | Vim
- Cloud Computing | Virtualization | Containerization | OOP | OOD | Full-Stack
- Change Management | Strategic Planning | Program Management | Product Design
- Multimedia Art | Cuisine and Cooking | English, Spanish

Projects

- **Cloud Infrastructure**: Providing assistance in a Cloud Computing research project at West Chester University. The project aims to develop a cloud infrastructure for high-performance computing systems using CI/CD pipelines (Virtualization, VMs, Containers, Kubernetes) (Spring 2023).
- **The Odin Project**: Built multiple small scale projects (Landing Page, Rock Paper Scissors, Etch-A-Sketch) During introductory HTML/CSS & Javascript self learning (JavaScript, HTML, CSS).
- **JavaScript 30**: Deployed 30 small scale JavaScript projects over 30 days as a coding challenge (JavaScript, HTML, CSS).
- **Cowbell:** Currently developing a web application to provide users with meal inspiration. This is a self learning/passion project with guidance from Jon Wexler's "Get Programming with Node.js" (Javascript, Node.js, MongoDB) **(12/2022 - Present).**

Education

Bachelor of Science — **West Chester University of Pennsylvania** — *West Chester, PA,* — **08/2020 - Present**
GPA 3.8/4

- Major in Computer Science; Minors in Spanish, Studio Arts
- Relevant Coursework: Computer Science III, Computer Systems, Discrete Mathematics, Cloud Computing, Modern Web Development
- Dean's List 2021, 2022; Computer Science Club

Experience

Coffee Shop Attendant — **Kendal-Crosslands** — *Kennett Square, PA* — **04/2018 - 08/2022**

Food Service Supervisor — **08/2022 - Present**

- Currently leading a team of 12 members in providing exemplary service to residents and staff.
- Evaluating staff skill and knowledge, providing consistent training and mentorship.
- Managing opening and closing sequences and providing recommendations to enhance efficiency in the workplace.
- Fostering positive attitudes, communication skills, and teamwork amongst team members.
- Reporting to weekly management meetings, providing feedback to upper management for staff needs.
- Assisting in meal operations in front and back of house.

Team Leader — **Maximo-Nivel** — *Cusco, Peru* — **07/2019 - 08/2019**

- Helped build and maintain a kindergarten during its mid-stage development.
- Provided guidance to teammates by communicating and translating with team management.