# Computers and Information Systems

Capstone Project (ECIS-690)

Submitted to:

Submitted by: Vijay Kumar Thati, Rahul Gargula

Spring 2024

Christian Brothers University



### **ABSTRACT**

The abstract talks about DevOps, a way of working together on software that makes it easier to deliver. It's about teamwork between the people who make the software (Dev) and the ones who run it (Ops). DevOps is all about working together, using machines to help, and using new ways to make software faster and better. It includes things like always adding new parts of the software, treating computer setups like code to make them easier to copy, and using special tools like Git, Bitbucket, Docker, Kubernetes, Argo CD to check the software works right. When companies use DevOps, they can get their software out faster, make it better, work together better, and find problems quickly. It's like breaking down walls between different teams so everyone can make better software together. But to do DevOps, companies need to change how they work and keep learning to do it better. Companies need to use DevOps because it helps them keep up with other companies in the digital world.

### **Introduction:**

Currently, as technology advances, there is a growing demand for automation in workflow and performance. Agile emphasizes communication between customers and developers to bridge gaps and achieve shared goals smoothly. DevOps seems to bridge the gap between developers and information technology infrastructure and operations.

CI/CD is a way of doing things with software that involves checking and using new code automatically. In this project, we want to make a way of turning code into real-life software that's

easy and fast. This means when we update the software, it happens quickly and works well right away. We worked on a Python application, so we focused on testing and using Python-based Django software. Also, we used orchestration tool something called Kubernetes to manage and grow the software.

Kubernetes High-Availability makes sure that if one part of Kubernetes stops working, everything else keeps going. Instead of just one boss controlling everything, we have lots of bosses to keep things running smoothly. Our project is about making a better way to put software out there using a CI/CD pipeline for a Python application. Our goal is to make the process quicker, more reliable, and easier to use and scalable on fly which will make the software better over time.

#### **TOOLS USED**

In DevOps, we use different tools and technology to help us with our work. These tools work with various infrastructures and platforms in the cloud. Here are the tools we used in our research:

**Docker:** Docker containers wrap software in a complete system that has everything needed to run the code. This means the software will always run the same, no matter where it's used or developed.

**Kubernetes:** Kubernetes is a tool that makes it easier to develop, scale, and handle applications that are stored in containers. It groups containers into small units for easy management and can manage clusters of hosts running Linux containers.

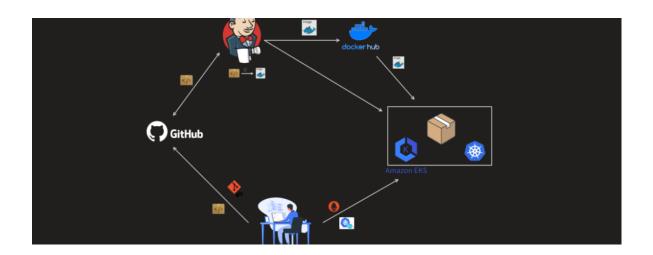
**Jenkins:** Jenkins is a tool for Continuous Integration. It helps build and test software projects continuously. This makes it easier for developers to make changes to the project and find out where the code isn't working. It also helps users get fresh builds easily.

**GitHub Actions:** This is a tool on GitHub that helps make development work easier and faster. It lets you customize and automate the steps you need to take when working on a project. For example, you can set it up to automatically test your code or deploy it to a website whenever you make changes. It's like having a helper that takes care of repetitive tasks for you.

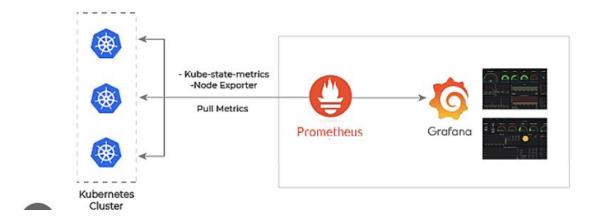
**Django Python:** Django is a type of programming language that's used to build web applications. It's like a set of tools that developers can use to create websites. Python is the main language used in Django, and it's known for being easy to read and understand. So, when we say "Django

Python," we're talking about using Django to build our website for the graduate application. This website will showcase the recent graduates from CBU college.

### IMPLEMENTATION OF THE SYSTEM



We've planned for how our project will run smoothly using a system called K8s. This system helps us manage big applications easily. Without needing people to do it manually, the computer will put our code onto the right platforms. We've set it up so that when we want to add more of our program to K8s, it can happen quickly and without causing any interruptions. And when we need less of it, we can shrink it down just as easily. This makes sure our program is always available when people need it, and it can handle lots of users without any problems.



## **GitHub Repository**

We made a special tool using Django for college students who are about to graduate. It lets users look up information about students who have graduated, and the results are managed by an administrator. Below, you can see the code for this tool.

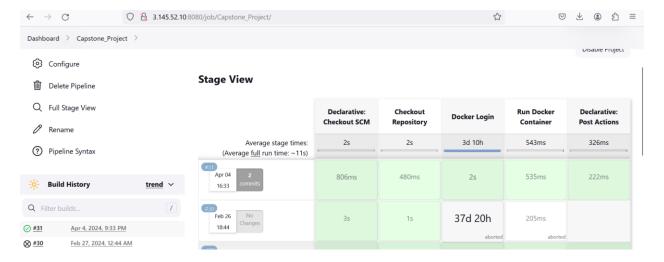
https://github.com/rahulcbu/Gradcap.git

😡 rahulcbu Update Dockerfile		57c7d34 · 3 weeks ago 39 Commits
idea .idea	Second commit	3 months ago
employees	First Commit	2 months ago
static	las	3 months ago
webApp	First Commit	2 months ago
Dockerfile	Update Dockerfile	3 weeks ago
JenkinsFile	Update JenkinsFile	2 months ago
LICENSE	Second commit	3 months ago
Pod	Create Pod	last month
README.md	Update README.md	2 months ago
db.sqlite3	First Commit	2 months ago
manage.py	Second commit	3 months ago

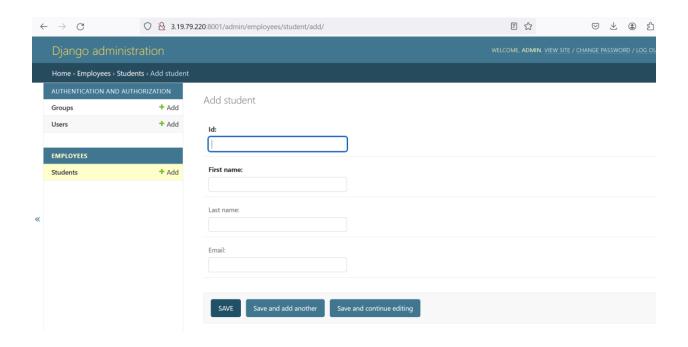
## Gradcap / Dockerfile ☐



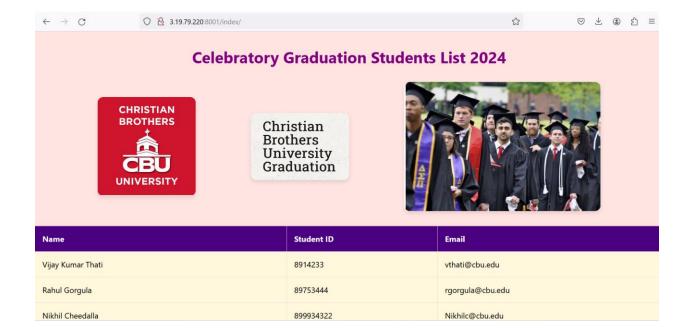
## **Pipelines configuration:**



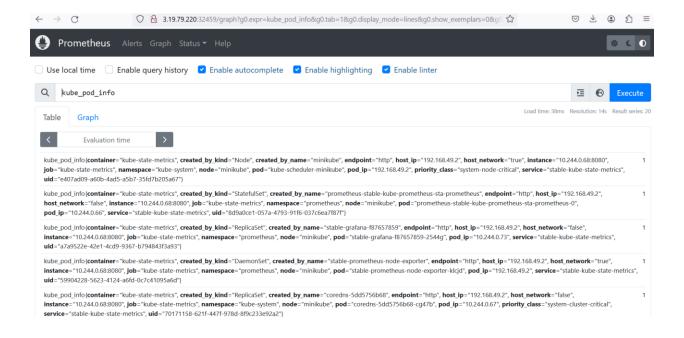
**Admin Page:** This page is used to give permissions to users who can have admin access and can able to edit the student list.



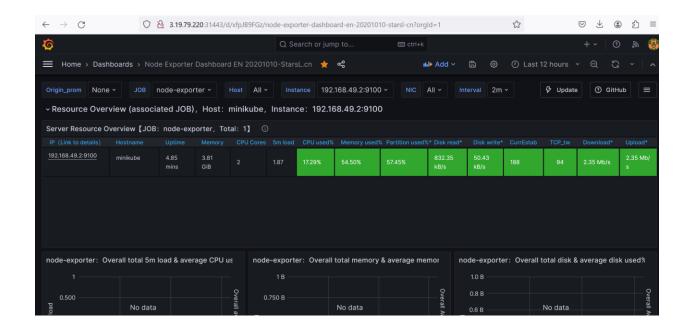
**Index Page:** This page shows the number of students that are graduating for summer 2024.



**Prometheus:** The below screen shot show the alert configuration for our servers.



**Grafana:** It is graphical representation/ data visualization used to see the details of the servers and their features.



### **Challenges Faced:**

Connectivity Problems: At first, connecting GitHub with Jenkins for continuous integration was tricky. To fix it, I had to make sure the right login details were used and double-check that the settings in GitHub matched those in Jenkins.

**Trouble with Docker Builds**: Sometimes, building Docker images in the pipeline didn't work due to missing parts or setup issues. I had to check the way the Docker file was written, make sure everything was set up right, and add any missing bits.

**Issues with Kubernetes Deployment**: When I tried to put applications onto Kubernetes clusters, I ran into problems with things like not enough resources or mix-ups in settings. Fixing this meant going through the setup files carefully, making sure everything was in the right place, and checking the logs for any details about what went wrong.

**Enhancing Pipeline:** As the project got bigger, the pipeline couldn't handle it as well. Buildings took longer, or we ran out of resources. I had to tweak Jenkins settings, make Docker setups bigger, and set up ways to save time, like caching, to speed things up.

**Monitor Pipeline:** I needed to watch the pipeline closely for any issues like builds failing or deployments not working right. To fix these, I added ways to log what was happening, set up tools like Prometheus and Grafana to watch how things were going, and used Jenkins plugins to get alerts if something went wrong.

#### **Conclusion:**

In conclusion, this project has accomplished several key milestones. Firstly, a GitHub account was created to manage project repositories, facilitating collaboration and version control. Secondly, a Django Python application was developed to share the graduation results of May 2024 students, enhancing accessibility to important information. Thirdly, an AWS cloud account was set up to host project resources, ensuring scalability and reliability. Fourthly, an EC2 instance was established on AWS to provide computing resources for the project. Fifthly, Jenkins was installed and configured on the EC2 instance to automate build processes, streamlining development workflows. Sixthly, security measures were implemented by configuring security tokens for EC2, GitHub, and the web application, safeguarding sensitive data. Seventhly, the Django application was deployed using Docker on the EC2 instance, and the image was pushed to Docker Hub for distribution, facilitating easy deployment and management. Eighthly, Kubernetes was utilized to deploy the Docker container, ensuring the scalability and reliability of the application. Ninthly, Prometheus and Grafana were configured to monitor the status of pods and nodes within the Kubernetes cluster, providing valuable insights into system performance. Overall, this project demonstrates the effective utilization of modern DevOps tools and practices, particularly Kubernetes, to efficiently scale a Django application, showcasing the project's commitment to innovation and excellence in software development.

### **References:**

- Gene Kim, Jez Humble, Patrick Debois: The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win (2013).
- Scott Chacon 2014 Pro Git Ben Straub This is a classic book on using Git, the version control system powering GitHub.
- Paul Du Bois 2014 The Docker Book John Wiley & Sons this book provides a good foundation for understanding Docker concepts.
- Sam Newman 2019 Building Microservices with Kubernetes O'Reilly This book focuses on building microservices using Kubernetes.
- Git SCM Git SCM website: https://git-scm.com/
- Jenkins Jenkins website: https://www.jenkins.io/
- Docker Docker documentation: https://docs.docker.com/
- Kubernetes Kubernetes documentation: https://kubernetes.io/docs/home/
- Prometheus documentation: https://prometheus.io/docs/introduction/overview/
- Grafana Grafana documentation: https://grafana.com/docs/grafana/latest/