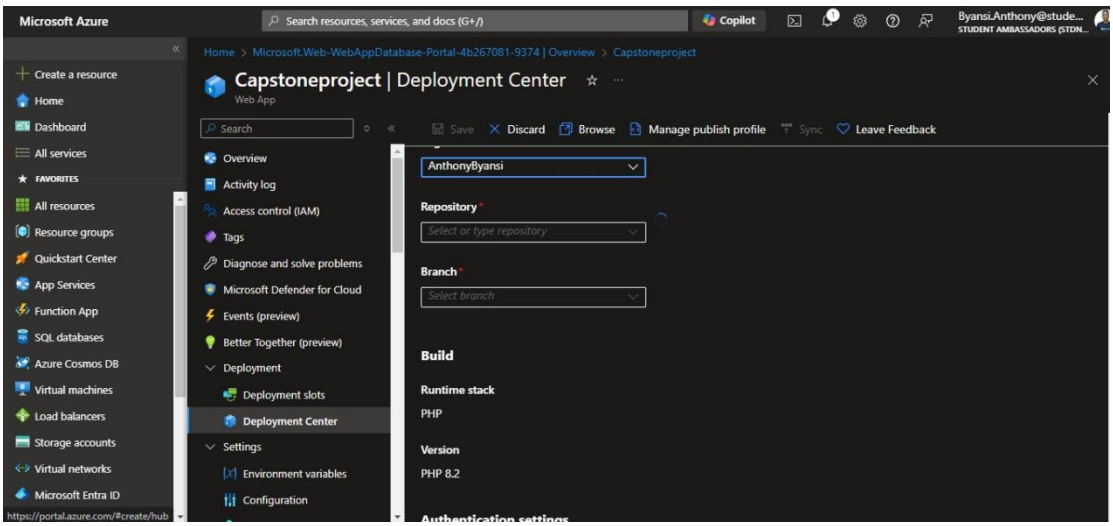
Week 3 Report: Continuous Deployment to Staging (Ending 4th Oct 2024)

OBJECTIVES

- **Set Up Deployment Scripts and Configure the Pipeline**: Establish and configure deployment scripts to ensure a seamless pipeline for deploying code to the staging environment.
- **Configure Cloud Provider (Azure)**: Set up and configure the cloud provider (Azure) to align with the needs of the staging environment.
- **Test and Verify Successful Staging Deployment**: Perform thorough testing and verification to ensure successful deployment in the staging environment, ensuring it replicates the production environment for reliable testing.

ACTIVITIES & ACCOMPLISHMENTS

Set Up Staging Environment:

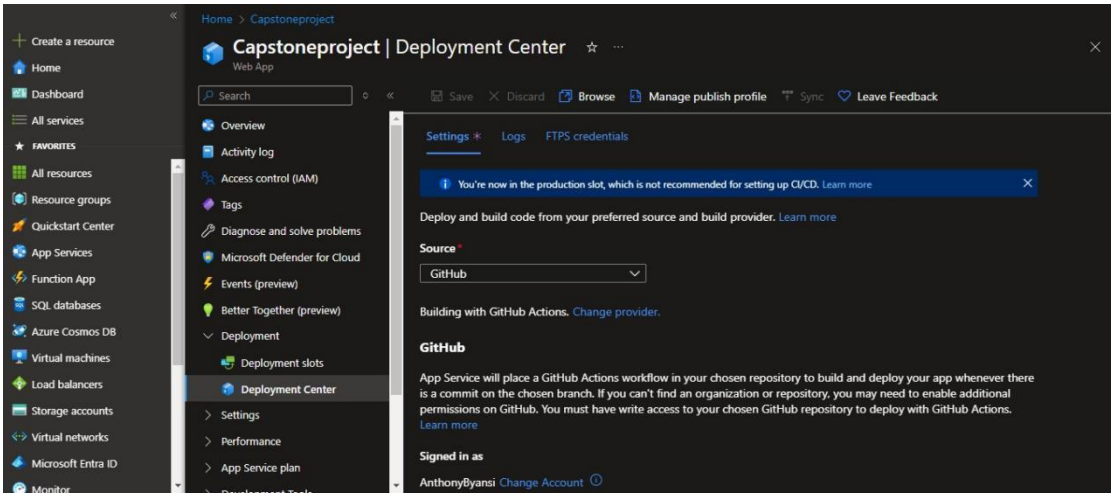


Cloud Service Provider: We chose to use the Microsoft Azure cloud platform as our cloud service provider due to its **robust security features**, **scalability**, **wide range of services**, **integration with Microsoft products**, and **support for hybrid cloud capabilities.**This was done through creating a new resource group in the Azure Portal to organize our resources.

Configuration: Configured the staging environment to mirror the production environment, including:

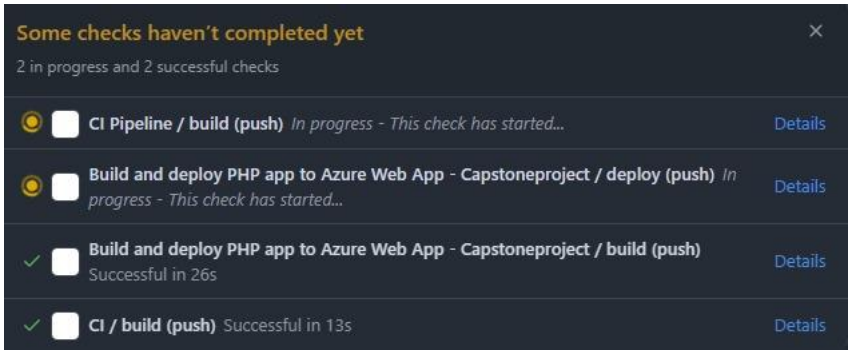
- **Database Setup**: Created a staging database (MYSQL) to ensure data integrity and accurate testing.
- **Environment Variables**: Set environment variables to match production settings, facilitating a consistent testing experience.
- **Networking Settings**: Configured networking settings to allow proper access and interactions among services. The WebApp was connected to MYSQL DB

Automate Deployment:



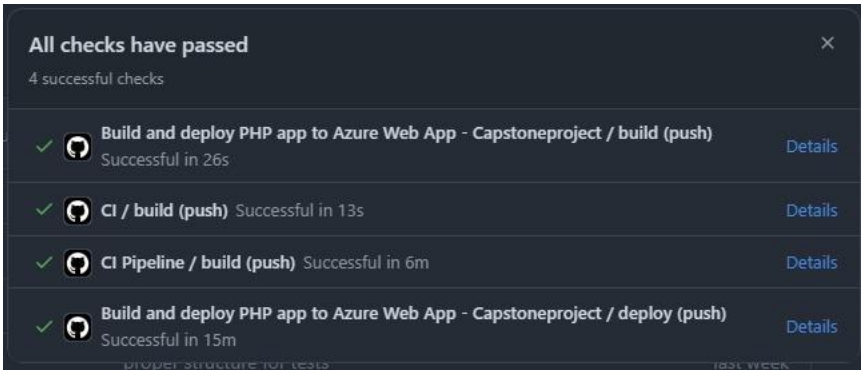
CI/CD Pipeline Configuration: Enhanced the CI/CD pipeline to automate deployment to the staging environment. Key features implemented include:

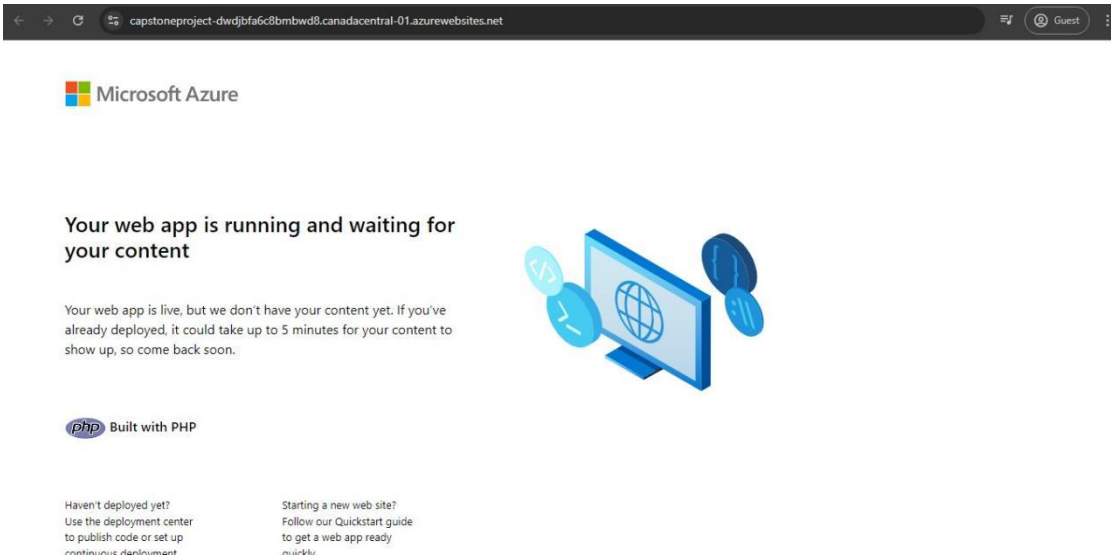
- **Trigger Mechanism**: The pipeline now automatically triggers deployment after successful execution of all tests.
- **Stages of Deployment**: Integrated stages for building, testing, and deploying the application, ensuring a robust deployment workflow.



Testing Deployment Process: Conducted thorough tests of the deployment process, resulting in successful initial test releases. Achievements include:

- Verified that all necessary components are correctly deployed in the staging environment.
- Ensured that the web application operates as expected, with all features functioning properly.

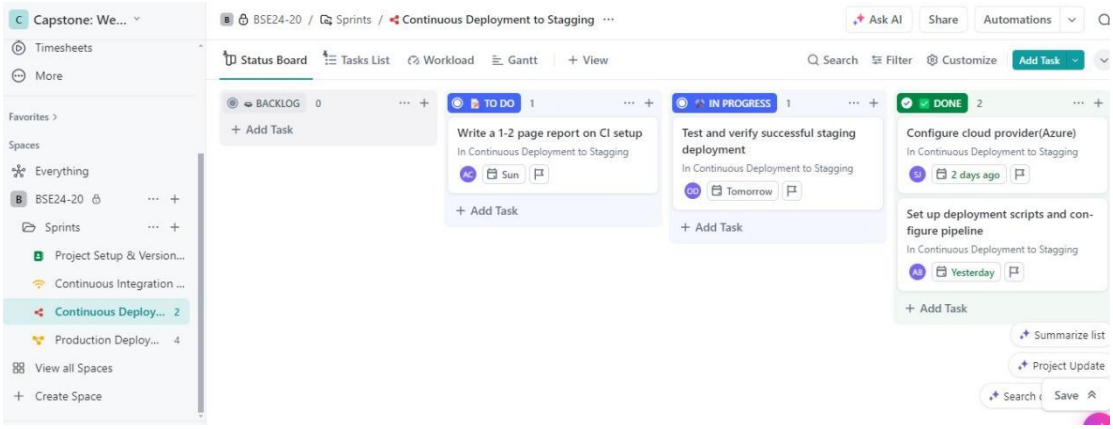




ClickUp Setup:

Task Assignment: Byansi Anthony created and assigned tasks in ClickUp for the following activities:

- **Setting up of Deployment scripts and configuring pipeline**: Assigned to Byansi Anthony.
- **Configuring cloud provider (Azure)**: Assigned to Sendi Joseph
- **Testing & Verifying successful staging deployment**: Assigned to Ochieng Devote Boniface
- **Writing a 1-2 page report on CI/CD setup:** Assigned to Aparo Cecilia



Deliverables

- **CI/CD Pipeline**: Established a fully functional CI/CD pipeline that automatically deploys to staging after passing all tests, significantly enhancing our development workflow.
- **Operational Staging Environment**: Successfully set up and launched an operational staging environment with the current version of the web application, allowing for further testing and validation.
- **Updated ClickUp Tasks**: All tasks in ClickUp have been updated to reflect progress, ensuring transparency and collaboration among team members.

Reflection on Challenges:

Below are some of the challenges we faced while staging the CD environment;

- Dependency Installation: running composer install is time-consuming, since there are many dependencies.
- Artifact Handling: Zipping and unzipping files is slow and resource-intensive(in the github actions consumption perspective)
- Azure Login: authenticating to Azure and deploying the app is slowed down by network latency and authentication issues.
- Resource Constraints: Limited computational resources can throttle deployments, particularly for large projects.