DOSR

DevOps Stack Reborn

Project by Tom Carrio and
Chase Ehlert

Project Idea

Our project this semester is similar to what we wanted to do last semester but more functional. Last semester we were not able to set up all of the services to be continuously integrated and one of them wasn’t working. This semester we set out to accomplish a set of functionality with features such as Team Collaboration, Continuous Integration, Single Sign-On, LDAP Integration, and more.

What All Is In It?

- GitLab CE
- GitLab CI
- **AWS Runner Cluster**
- Docker Containers
- Mattermost
- LDAP Authentication
- **Single Sign-On**
- 2 Factor Authentication
- PostgreSQL
- Nginx
- Flask
- Gunicorn
- S3 Backups
- Glacier Migration
- SMTP (Please make the build emails stop)
- Webhooks

Team Roles

Tom:

- Python
- AWS EC2 Cluster
- GitLab
- Mattermost
- Webhooks
- Nginx
- Docker
- SSO
- LDAP
- Web-backend
- Cloud Backups

Chase:

- ERD Designer
- Relational Schema Designer
- Database Administrator
- Security Analysis
- SMTP
- Team Motivator

GitLab CE

GitLab is a version control software that uses git (go figure) for the backend and offers multiple tools such as issues, pull requests, Git repository management, code review and more. GitLab CE, Community Edition, is the free and open-source component of the complete Enterprise Edition, which still offers many services which will be shown.

For reference: GitHub

GitLab CI

The integration of GitLab CI with our system allows for the availability of building configurations to test the code in every commit, pull request, merge, and more. This will let a team quickly confirm working code while minimizing risk. The build executor can be set to many different configurations, but the Docker executor was our preferred choice and was configured for containerized building of projects. The docker-in-docker functionality was also tested and enabled.

Mattermost

Mattermost is a messaging system oriented for project development, utilizing teams and channels for discussion. Mattermost can also be configured with incoming/outgoing webhooks for external service, allowing integrations such as issue notification and project deployment from a chat channel. This is our **community interaction component** that makes catching up with your dev team quick and easy.

PostgreSQL

PostgreSQL is an object-relational database. It is used for all services offered, with databases backing our website, GitLab, and Mattermost. Similar to MySQL, it is free and open source.

Website

Due to the minimal size of the website, it was developed using the Flask micro web framework and integrates with our database using the psycopg2 module. A systemd service unit was created for launching the application using gunicorn, and the website is served with Nginx using a unix socket proxy pass.

Backup Plan

AWS was utilized for our backup strategy. The PostgreSQL databases used by our applications are dumped to SQL files with the pg_dump tool. Afterwards, the aws-cli utility is used in a backup script for pushing new versions of configuration and SQL files to an S3 bucket. The S3 bucket is history versioned, and configured to migrate old versions of files in the bucket to Glacier after 14 days and delete off S3 entirely after 30.

Demonstration

DOSR

http://dosr.carrio.me

These are single core virtual machines. Plz don’t be a DDOSr.

Q & A

Sample Questions:

- Why didn’t you guys host a videogame on your server?
- Whose idea was this?
- What’s your favorite color?
- How much wood can a woodchuck chuck if a woodchuck could chuck wood?