CPSC 319
Zenith Blog
Installation Guide

Presented By:

Muhammad Saad Shahid
Param Tully
Eric Wong
Cheryl Yu
Anusha Saleem
Shawn Zhu
Andrew Liu
Ruchir Malik
Anthony Baek

Sponsors:

Evan Seabrook
Alvin Madar

TA:

Asem Ghaleb

Version: 1.0
Date: Apr/2/2023

INDEX

Steps to Setup Local Environment..... 3
Step 1 - Homebrew:..... 3
Step 2 - Eclipse Temurin:..... 3
Step 3 - NodeJS:..... 3
Step 4 - Google Cloud SDK:..... 4
Step 5 - Docker:..... 4
Step 6 - NPM:..... 4
Step 7 - Maven:..... 4
Step 8 - SpringBoot CLI:..... 4
Step 9 - Authenticate:..... 4
Step 10 - Clone Github Repo:..... 5
Step 11 - Run Backend:..... 5
Step 12 - Run Frontend:..... 6
Step 13 - Download Slack:..... 7
Step 14 - Run Backend Tests:..... 8
CI/CD Setup for Google Cloud QA & Prod Environment..... 9
Overview of Google Cloud CI/CD setup:..... 9

Steps to Setup Local Environment

Before you start working on the project, install the following services/software on your computer:

1. A web browser. Ideally a Chromium-based such as Google Chrome or Microsoft Edge
2. A terminal environment, since this guide is written from the perspective of a MacOS user using zsh shell. We would extensively use the Homebrew package manager so Linux users can also follow the same commands

Step 1 - Homebrew:

Install the Homebrew package manager by running the following command in the terminal:

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Step 2 - Eclipse Temurin:

Install Eclipse Temurin Java Development Kit by running the following command in the terminal

```
brew install --cask temurin
```

Step 3 - NodeJS:

Install NodeJS:

```
brew install node
```

Step 4 - Google Cloud SDK:

Install Google Cloud SDK:

```
brew install --cask google-cloud-sdk
```

Step 5 - Docker:

Install Docker

```
brew install docker
```

Step 6 - NPM:

Install npm

```
install -g npm
```

Step 7 - Maven:

Install Maven

```
brew install maven
```

Step 8 - SpringBoot CLI:

Install SpringBoot CLI

```
brew tap spring-io/tap
brew install spring-boot
```

Step 9 - Authenticate:

Authenticate through Google Cloud CLI. You should use the following email address and password: Email: zenithblogteam@gmail.com
Password: cpsc319zenith

```
gcloud auth application-default login
```

Step 10 - Clone Github Repo:

Clone the GitHub repo at the desired location on your system:

```
git clone https://github.com/CPSC319-2022/zenith.git
```

Step 11 - Run Backend:

Once the repo is cloned, cd to blog directory from the root of the cloned repo and start the spring boot app:

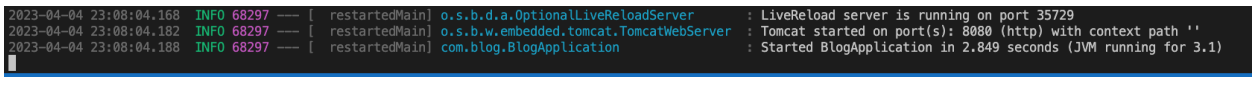
```
cd  blog
```

And typing:

```
./mvnw spring-boot:run
```

You should see the following:

The app is running when the following message is visible in the terminal:



Step 12 - Run Frontend:

Open another terminal window, and cd to the frontend directory (assuming that you are at the root of the repo):

```
cd frontend
```

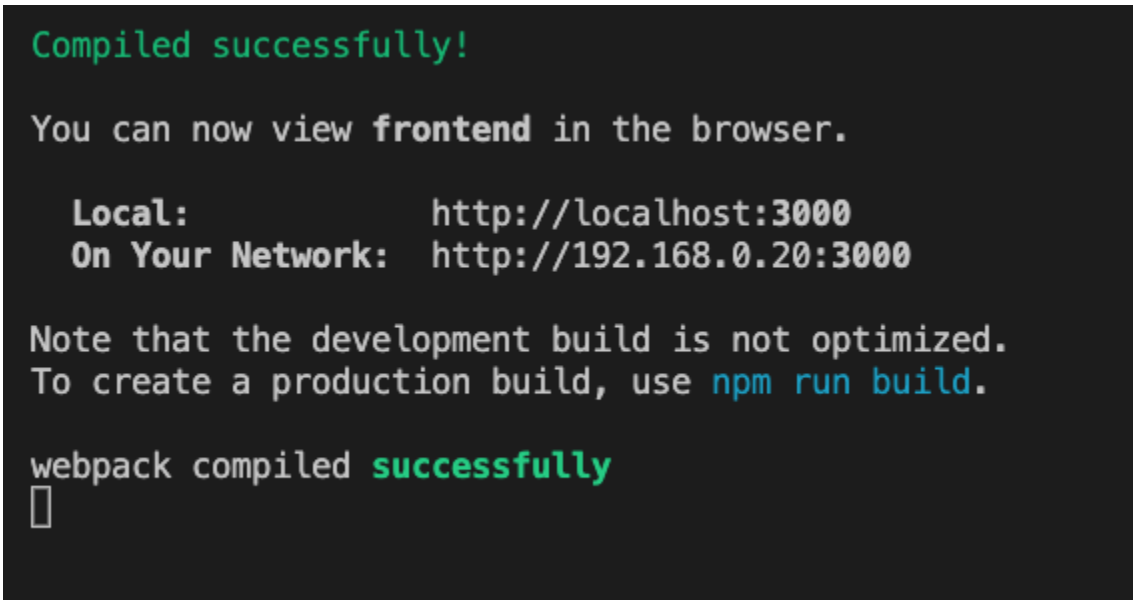
And type the following command to install relevant node packages:

```
npm install --legacy-peer-deps
```

After the node packages are installed, run the frontend ReactJS app by typing the following at ./frontend:

```
npm start
```

You should see the following:



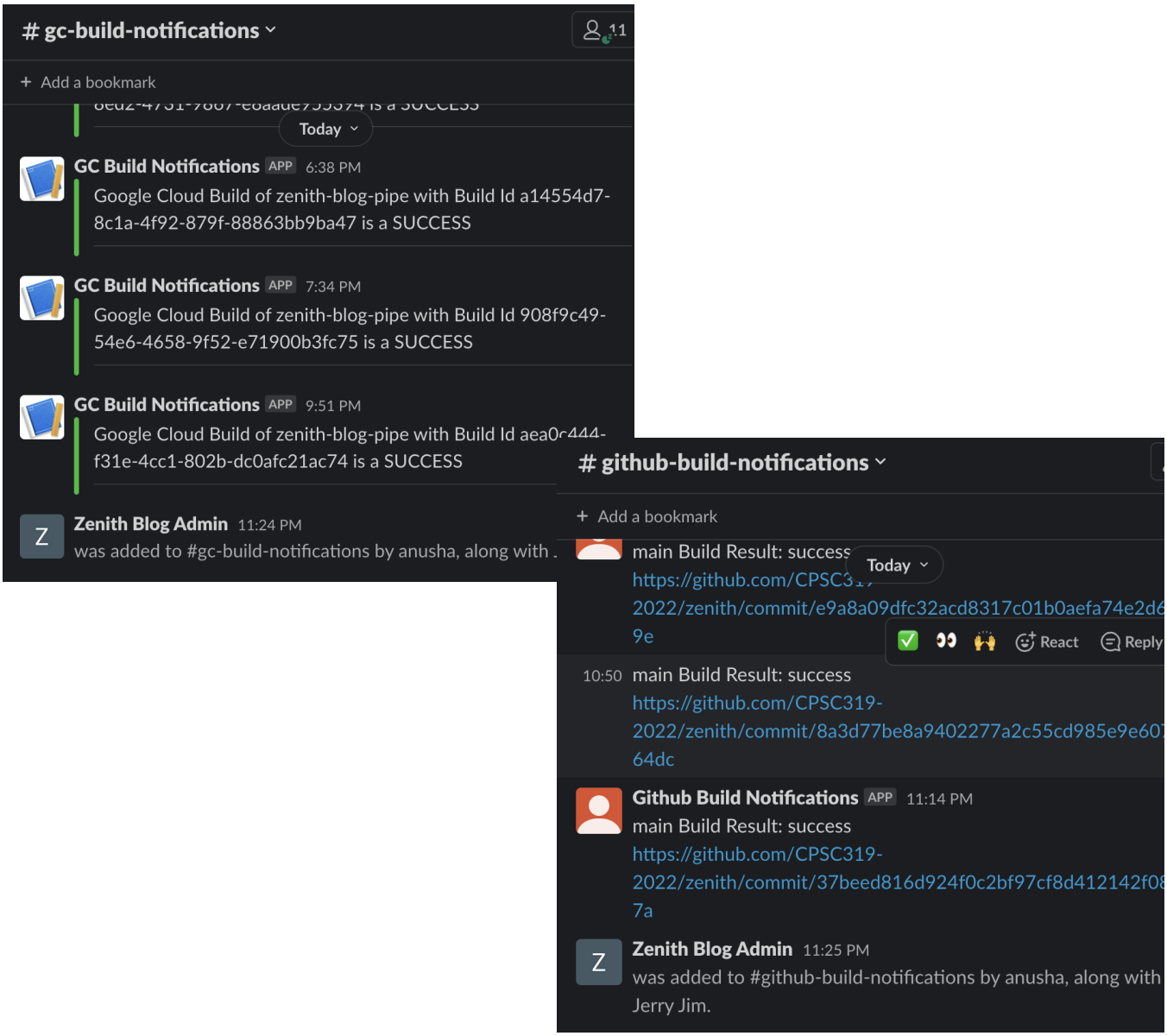
Congratulations! Now you are all setup for local development.

Step 13 - Download Slack:

A crucial part of this blog application is to receive build notifications on Slack. As an Admin with the following credentials (zenithblogteam@gmail.com, cpsc319zenith), you are already added to the Zenith Blog Slack Workspace and the two channels #gc-build-notifications and #github-build-notifications.

For anyone else, please use this invite link to download Slack and join our workspace:

```
https://join.slack.com/t/zenithblog/shared_invite/zt-1sfnfibti-WknS_2hid9oWMHn3g1PfRA
```



Step 14 - Run Backend Tests:

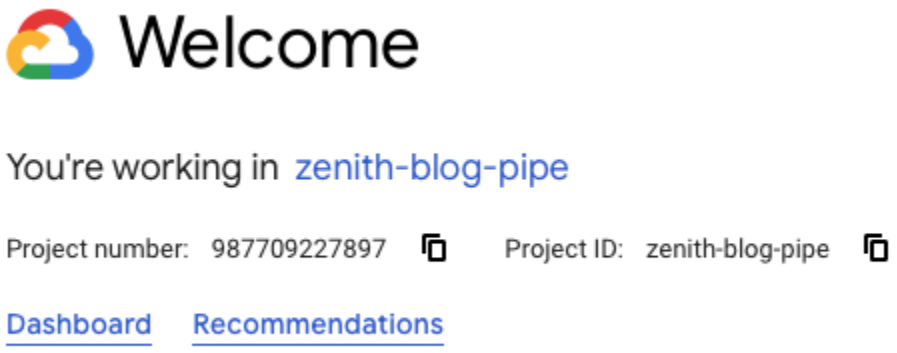
To run the tests, please navigate to the blog folder, and type in the following command

```
./mvnw test
```

CI/CD Setup for Google Cloud QA & Prod Environment

Overview of Google Cloud CI/CD setup:

Our project is build and deployed on the web using Google Cloud. You should use the admin email address that shared earlier as it has owner privileges already setup.
https://console.cloud.google.com/welcome?project=zenith-blog-pipe



In the repo, there is a cloudbuild.yml file at the root which serves as a configuration file for google cloud build.

The cloudbuild.yml file refers to the frontend dockerfile (located in the ./frontend directory) and Dockerfile at (located in the ./blog directory) to build docker and run docker containers that are used for cloud build and deployment.

The Cloud build triggers are set on every push on the ‘prod’ and ‘qa’ branches. The approach is best represented by the following diagram:

We have the following approach:
An automatic *Pull Request to QA on successfull push/pull to dev*
A manual pull request to prod from QA since QA has already preventing the failing code. A series of tests run on each push to a branch.

Also, as per project specs, pushes on non-feature branches are deployed, where as pushed to a feature branch are only subject to automated tests. We are using GitHub actions to accomplish this.

Following Google Services are being used:

1. Google Cloud Build
2. Google Run to deploy the production app and testing/qa sandbox.
3. Google OAuth 2.0 for user authentication
4. Pub/Sub service for providing build notifications to Slackbot and WebUI
5. Google Artifact registry for hosting images
6. Google Storage bucket
7. Cloud SQL as a database solution

To add a new user to the project, login to the google cloud console and follow the steps listed here: https://cloud.google.com/iam/docs/grant-role-console

To see build status, go to
https://console.cloud.google.com/cloud-build/builds?referrer=search&project=zenith-blog-pipe

The deployed services can be found at
https://console.cloud.google.com/run?referrer=search&project=zenith-blog-pipe

The domain mapping for https://zenithblog.ca is linked with the prod run url: https://console.cloud.google.com/run/detail/us-central1/zenith-prod/metrics?project=zenith-blog-pipe