DevOps and Cloud Research

Daniel R. Cender

Grand Canyon University: CST-323-O500

Dec 6, 2020

**Test Application**

**GitHub URL:** [**https://github.com/DanielCender/CST-323-Test-App**](https://github.com/DanielCender/CST-323-Test-App)

**Deployment URL:** [**Cst323TestApp-env.eba-uvqpnjq2.us-east-1.elasticbeanstalk.com**](http://cst323testapp-env.eba-uvqpnjq2.us-east-1.elasticbeanstalk.com/)

The test application for this course activity set is still in development, though significantly further along than in the last milestone. It will be a multi-page lite-blogging application (think of Twitter more than WordPress), connected to a MySQL database. It’s being developed using NodeJS and the React framework, with Bootstrap for a consistent style system.

Users will “claim” a username for the time they are using the app. After they logout, that name will again be available for others to write posts under (or delete old ones). The main feed page has a basic text editor form and a scrolling view of the latest posts on the site. By clicking on the user’s username near the Logout button, they navigate to a view where they can manipulate all their user’s posts.

**ER Diagram**

As of this milestone, the test application requires only one database table to function.

Graphical user interface, application

Description automatically generated

**UI Designs**

Some basic preliminary designs made in Canva.

Graphical user interface, application, Teams

Description automatically generated

Graphical user interface, text

Description automatically generated

A picture containing graphical user interface, text

Description automatically generated

There will be an “About” page, including a feedback form that will create a comment in a different database table.

**Service API Design**

The API servicing this small application will have only a handful of routes for manipulating and querying the Post records from the connected database.

* /api/posts (GET)
* /api/post/:id (GET)
* /api/post/create (POST)
* /api/post/update (POST)

**Screenshots**

Graphical user interface, text, application, email

Description automatically generated

**Logging**

Error tracking and notification integrations are being handled by Sentry, which has an easy plugin for React. A screenshot of this application’s dashboard is below:

Graphical user interface, text, application, email

Description automatically generated

**Uptime Tracking**

Site uptime and outage notifications are being handled by Uptime Robot. It’s used to verify a web service is basically functional and available on specified ports or protocols. Below is this app’s dashboard:

A picture containing text, screenshot, monitor

Description automatically generated

Cloud Computing Research

Nothing for most of the research questions.

**Three Tools to Support CI/CD Build Pipeline**

A few tools that are used by many teams to implement CI/CD:

1. Jenkins - This is an open-source automation server that hosts a CI/CD process. It works on Windows, MacOS, and Unix-based systems. It relies on a sizable library of plugins which allow the system to build a variety of software projects.
2. CircleCI - This system integrates heavily with GitHub, GitLab, and BitBucket repositories to automate testing and deployment, both on a cloud-hosted option or private infrastructure. It boasts a quick setup and great support for automated testing.
3. Buddy - Much like CircleCI, this system deploys code from GitHub, GitLab, or Bitbucket repositories. It uses Docker containers with pre-installed languages and framework tools, which include those for status notifications and monitoring, to build software. Like Heroku, this service offers attachments to 3rd party services like MariaDB, Elastic, PostgreSQL, RabbitMQ, Redis, and some others.

**Five Capabilities That Drive DevOps**

1. Automate Infrastructure

The process of deploying code in a smooth system is only possible if the infrastructure runs on is consistently dependable, as far as its configuration. Plenty of these options are available now through IaaS and PaaS solutions.

1. Automate Deployments

Automating deployments involves connecting the deployment process to the team's source control service or repository. These forms of automation usually pick up small, incremental changes to the main source control branch (and peripheral feature or test branches) and run the new source through the CI/CD pipeline for its respective environment. This ensures that breakages are found faster and addressed speedily. There's likely a lot of less finger-pointing and group debugging sessions when it's easier to know who and what code broke the build.

1. Design For Feature Flags

This is a method of designing applications and their deployment in such a way that it's almost modular. If this is done properly, then a production build can be configured to only allow certain features to be accessible for specific user groups, in order to test smaller features before mass-distributing them. A modern-day example of this might be when Instagram removed it's "like" count from peoples' posts, but only for specific user regions.

1. Measure, Monitor, Experiment

In a well-rounded DevOps environment, systems will be in place that allows developers to monitor different portions of the application for insights into bugs or potential future fixes. These sorts of monitoring services work in conjunction with things like feature flags to make sure developers can "fail fast" and iterate quickly through designs.

1. Continuous Integration and Continuous Delivery

This is what most people think is primarily involved in DevOps. CI relates to testing the code contributed by developers to the central codebase repository system. CD relates to grabbing that source code and transforming it into a deployable package for consumption by the end user.

Many of the benefits of a great DevOps environment have become reachable for teams of all sizes and maturities, largely in part to the widespread availability of the cloud and PaaS/IaaS services.

References

Kavis, M. (2014). *Architecting the cloud: Design decisions for cloud computing service models (SaaS, PaaS, and IaaS)*. Hoboken, NJ: Wiley.