Continuous Deployment

17-313 Fall 2025
Foundations of Software Engineering
https://cmu-17313q.github.io
Eduardo Feo Flushing

S3D

Continuous Delivery: Why?

"The biggest risk to any software effort is that you end up building something that isn't useful. The earlier and more frequently you get working software in front of real users, the quicker you get feedback to find out how valuable it really is."

Martin Fowler, Continuous Delivery

Knight

Knight Capital posts $389.9 million loss on trading glitch

Knight Capital Group reported a third-quarter net loss due to a glitch on August 1 that forced the electronic trader to take on additional investors to avoid...

Oct 17, 2012

Knight Shows How to Lose $440 Million in 30 Minutes

Talk about a bad day. In the mother of all computer glitches, market-making firm Knight Capital Group lost $440 million in 30 minutes on Aug...

Aug 2, 2012

Motivating scenario: Failed Deployment at Knight Capital

"It took 17 years of dedicated work to build Knight Capital Group into one of the leading trading houses on Wall Street. And it all nearly ended in less than one hour."

Knightmare: A DevOps Cautionary Tale

👤 D7   📁 DevOps   🕐 April 17, 2014   ≡ 6 Minutes

I was speaking at a conference last year on the topics of DevOps, Configuration as Code, and Continuous Delivery and used the following story to demonstrate the importance making deployments fully automated and repeatable as part of a DevOps/Continuous Delivery initiative. Since that conference I have been asked by several people to share the story through my blog. This story is true – this really happened. This is my telling of the story based on what I have read (I was not involved in this).

This is the story of how a company with nearly $400 million in assets went bankr[upt in] minutes because of a failed deployment.

"In the week before go-live, a Knight engineer manually deployed the new RLP code in SMARS to its 8 servers. However, he **made a mistake and did not copy the new code to one of the servers.** Knight did not have a second engineer review the deployment, and neither was there an automated system to alert anyone to the discrepancy. "
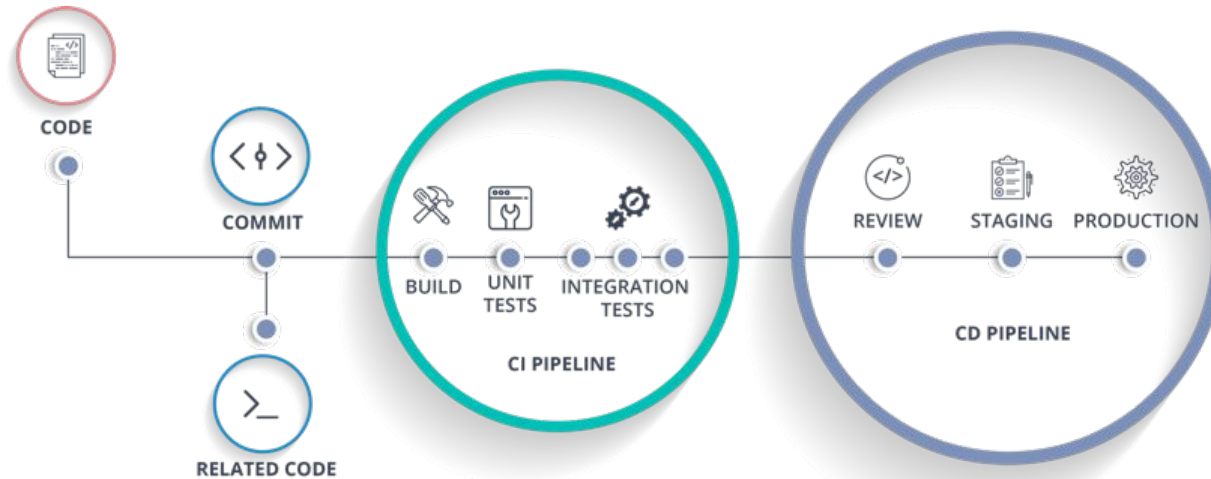
Mellon University

What could Knight capital have done better?

- Avoid including "test" and "dead" code in production deployments

- Automate deployments

- Define and monitor risk-based KPIs

- Create checklists for responding to incidents **(Risk Management!)**

CI/CD Pipeline overview
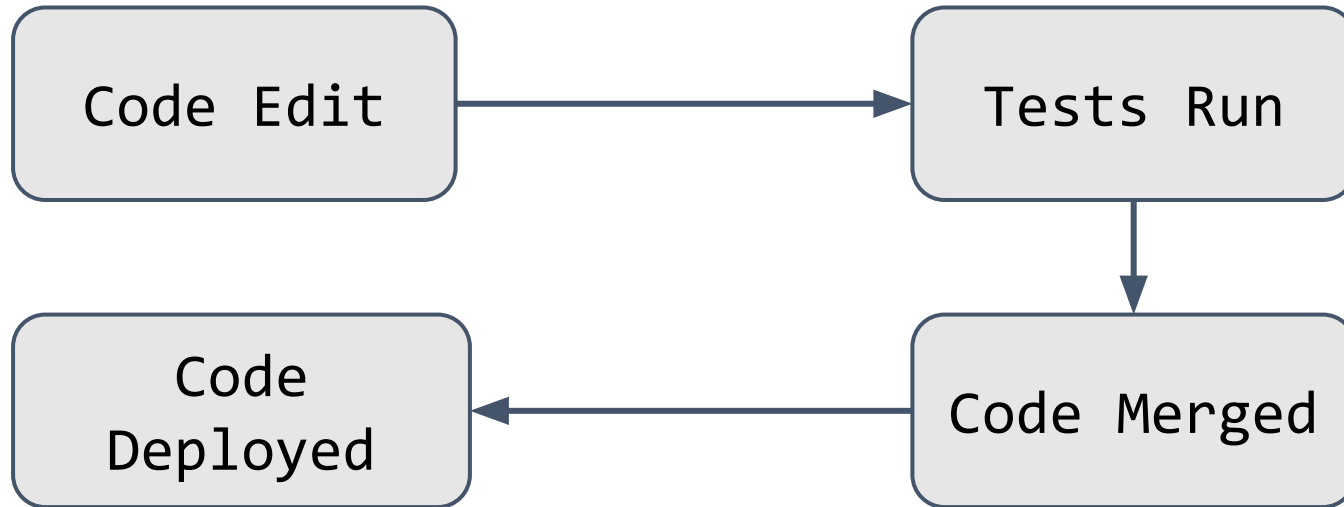
Example CI/CD Workflow

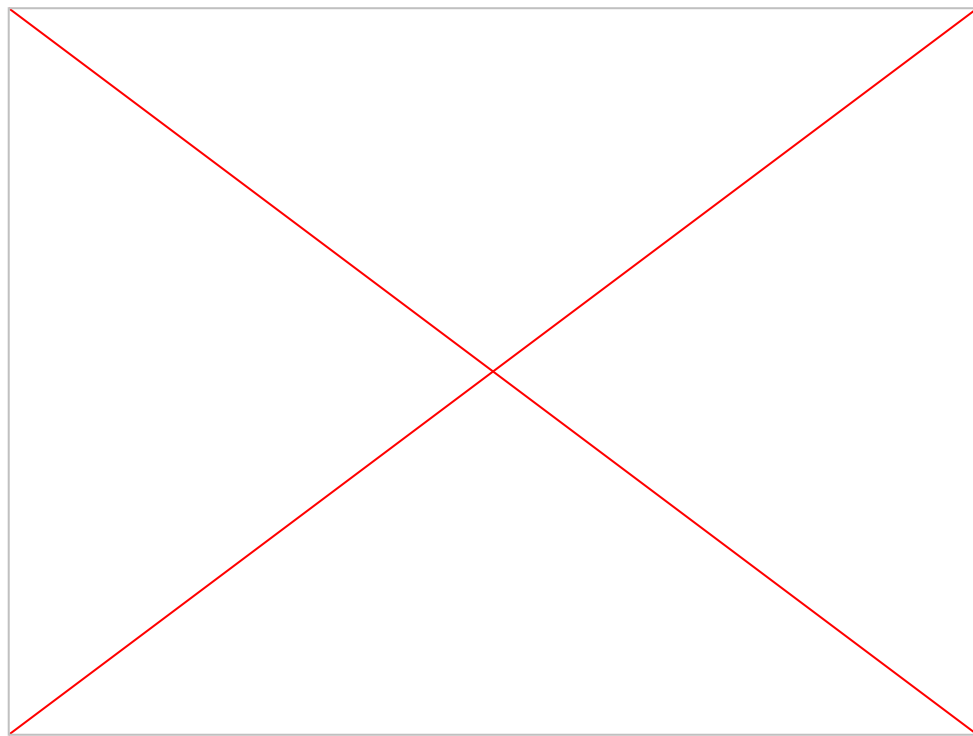# Continuous Delivery != Immediate Delivery

- Even if you are deploying every day ("continuously"), you still have some latency

- A new feature I develop today won't be released today

- But, a new feature I develop today can begin the **release pipeline** today (minimizes risk)

- Release Engineer: gatekeeper who decides when something is ready to go out, oversees the actual deployment process

S3D

Carnegie
Mellon
University

How can we continuously deploy our software in production?

Continuous Deployment / Continuous Delivery
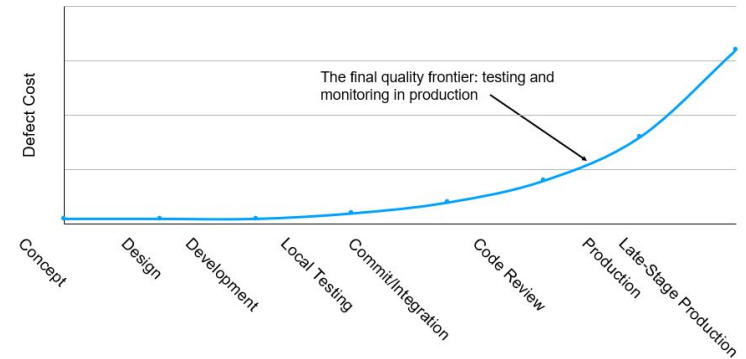
Carnegie
Mellon
University

Continuous Delivery /Deployment
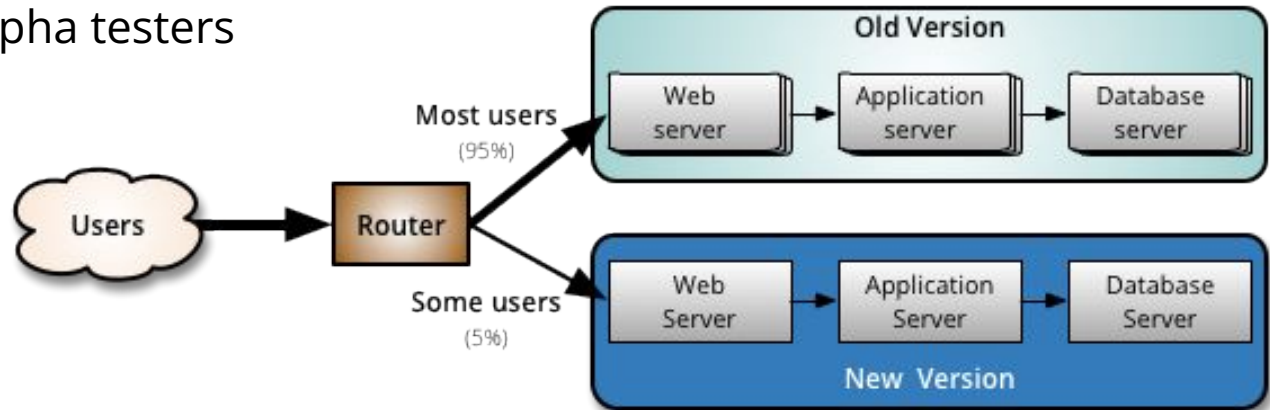
Done right

Continuous Delivery

- "Faster is safer": Key values of continuous delivery
  - Release frequently, in small batches
  - Maintain key performance indicators to evaluate the impact of updates
  - Phase roll-outs
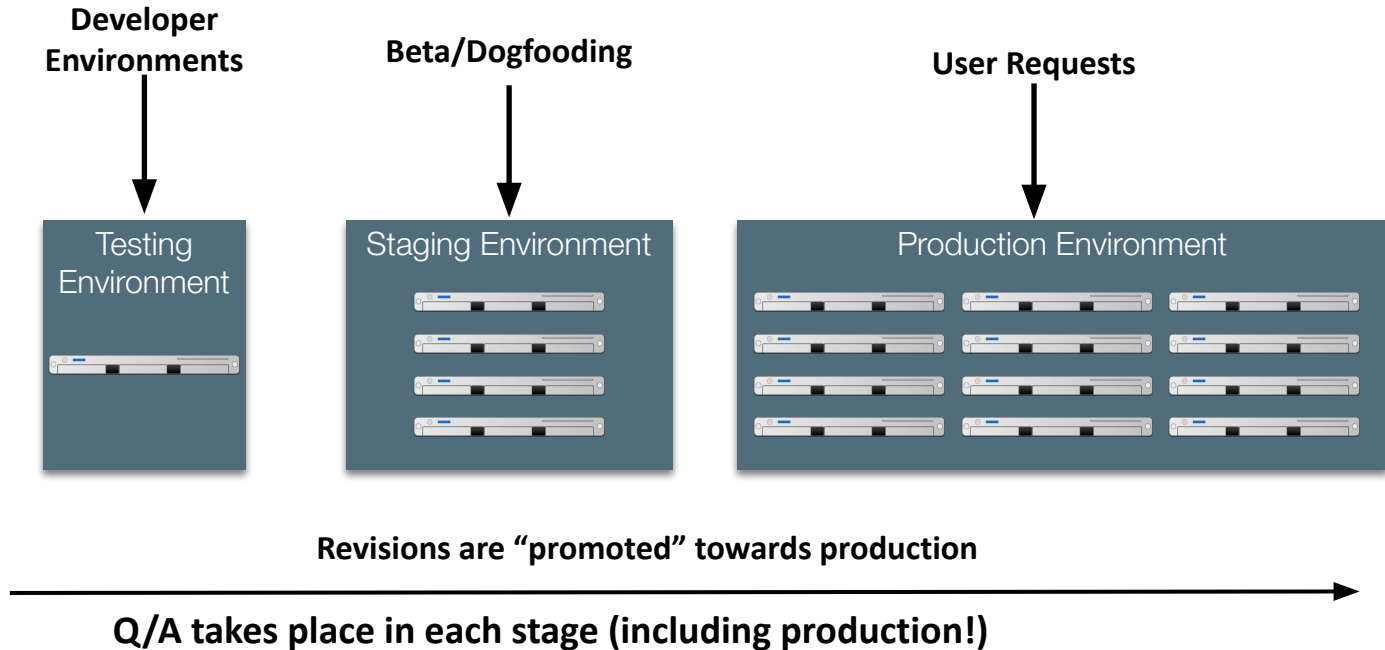 - Evaluate business impact of new features

Split Deployments Mitigate Risk

- Idea: Deploy to a complete production-like environment, but don't have users use it, collect preliminary feedback

- Lower risk if a problem occurs in staging than in production

- Examples:
  - "Dogfooding" "Eat your own dogfood"
  - Beta/Alpha testers

Staging Environments for Continuous Delivery

Developer Environments

Beta/Dogfooding

User Requests

Testing Environment

Staging Environment

Production Environment

Revisions are "promoted" towards production

**Q/A takes place in each stage (including production!)**
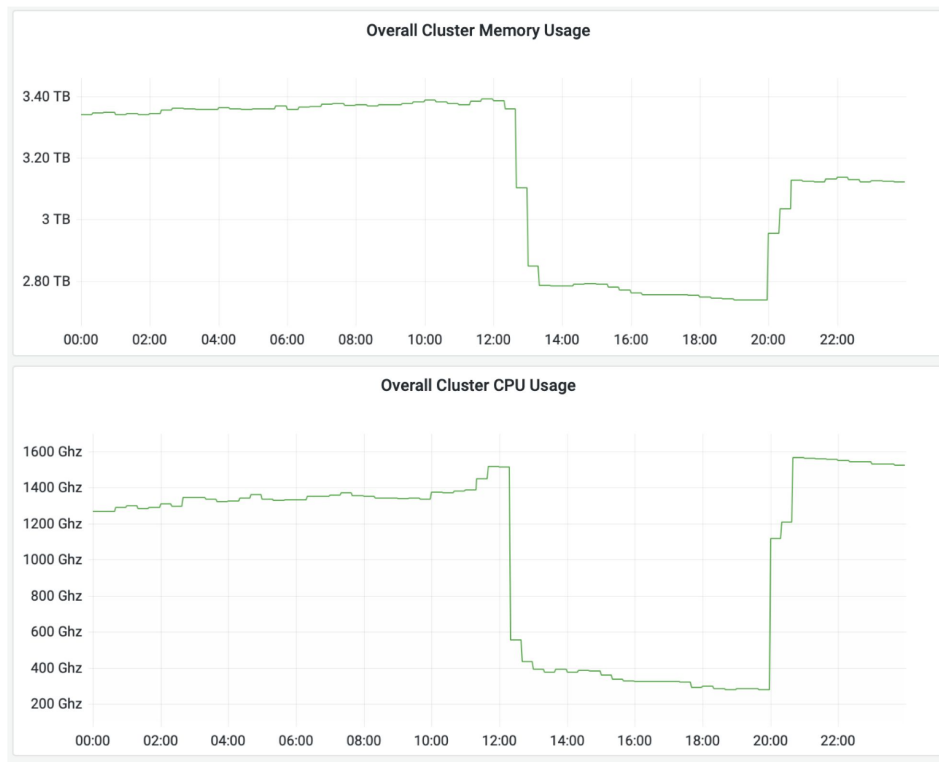
Continuous Delivery Tools

- Simplest tools deploy from a branch to a service (e.g. Vercel. Render.com, Heroku)

- More complex tools:
  - Auto-deploys from version control to a staging environment + promotes through release pipeline
  - Monitors key performance indicators to automatically take corrective actions

Example CD pipeline from Spinnaker's documentation:
https://spinnaker.io/docs/concepts/#application-deployment

Carnegie
Mellon
University

Continuous Delivery **Needs** Monitoring

- Consider both direct (e.g. business) metrics, and indirect (e.g. system) metrics

- Hardware

  - Voltages, temperatures, fan speeds, component health

- OS

  - Memory usage, swap usage, disk space, CPU load

- Middleware

  - Memory, thread/db connection pools, connections, response time

- Applications

  - Business transactions, conversion rate, status of 3rd party components, logs

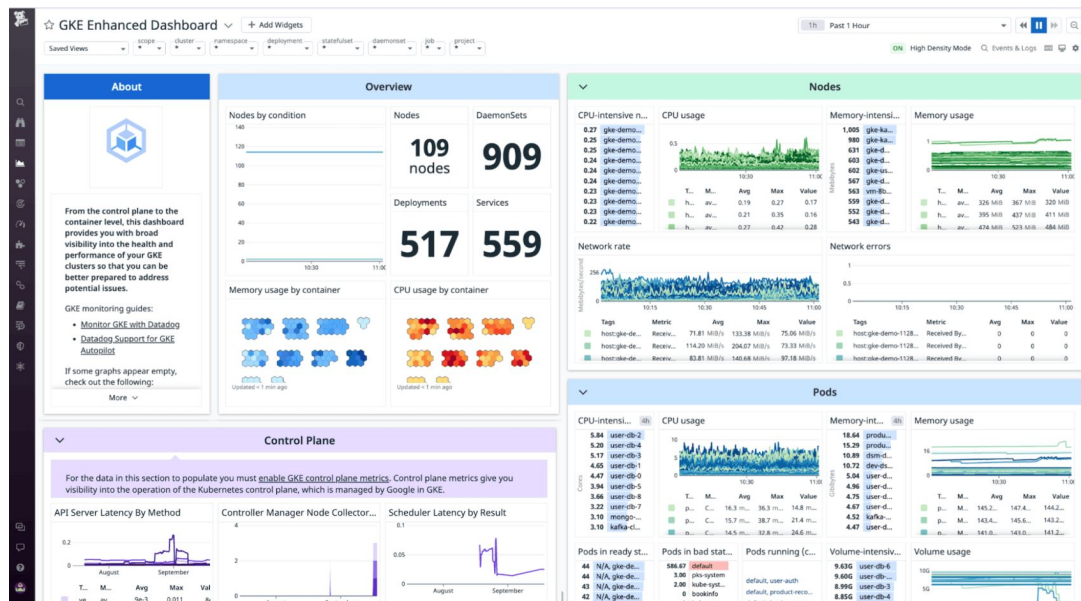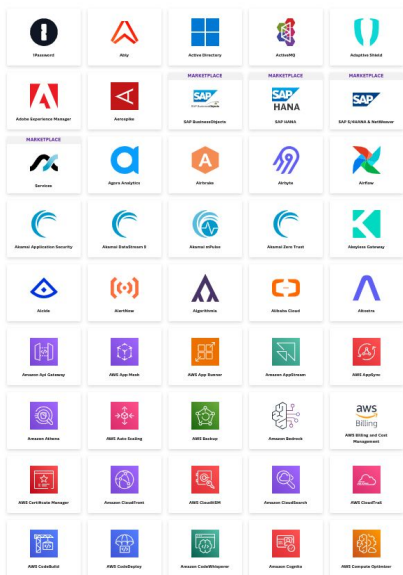# Monitoring can help identify operational issues
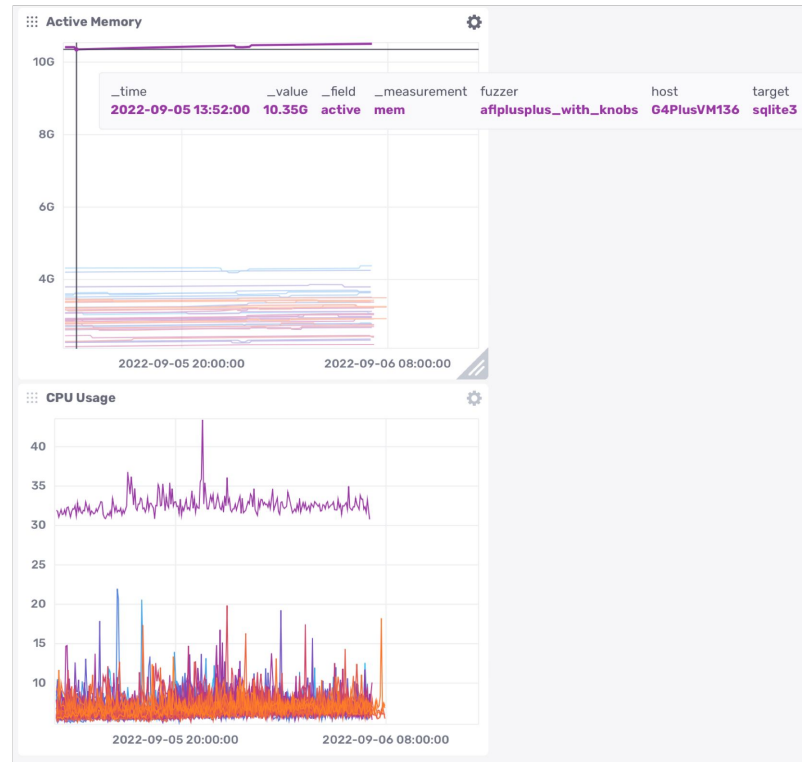
Cloud Monitoring as a Service
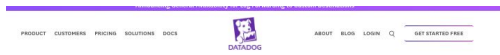
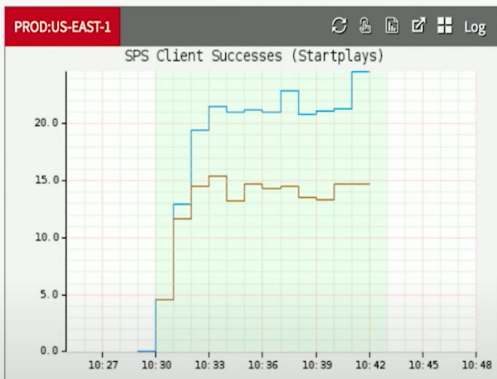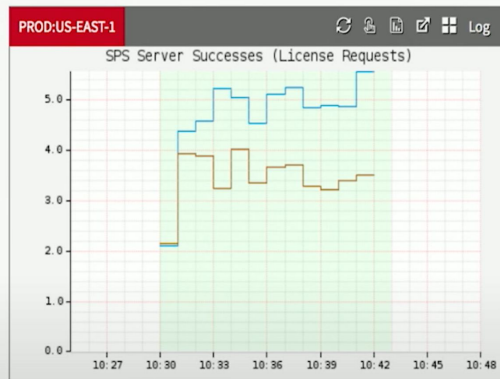# Continuous Delivery Tools Take Automated Actions

- Example: Automated **roll-back** of updates at Netflix based on SPS

Activity: Try CD by yourself

1. Fork this repo:
   https://github.com/CMU-17313Q/basic-web-app-f25
2. Follow the instructions in the readme to run and test the development server locally.
3. Once you have it running locally visit `http://localhost:3000` and try different queries like *Who was Shakespeare?* and *What is your Andrew ID?*
4. Complete the activity tasks following the instructions in cmu-17313q.github.io/recitations/deployment-workshop-f25/
5. `Submit a link to the deployed site (link on Slack)`

Configure Project

Project Name

basic-web-app

Framework Preset

N Next.js ⌄

Root Directory

./ Edit

Congratulations!

You just deployed a new Project to Vercel.

Welcome!!

Please enter your query in the box below:

No Match

Production Deployment

The deployment that is available to your visitors.

Build Logs   Runtime Logs   ↺ Instant Rollback

Welcome!!
Please enter your query in the box below:

No Match

Deployment

basic-web-app-vjlw-59ccn2ml2-eduardos-projects-cb396bfb.vercel.app

Domains

basic-web-app-vjlw.vercel.app ↗ +2

Status   **Created** ⓘ

● Ready   2m ago by anudaweerasinghe

Source

⌥ main

⦿ 50fbed7 Merge pull request #5 from CMU-313/update-f23

S3D

Carnegie
Mellon
University