Tech Workshop #2

Braden Guliano

Director of Technical Development

Kappa Theta Pi

Prerequisites

GitHub account

IntelliJ WebStorm

Git

Node JS

Git for Windows

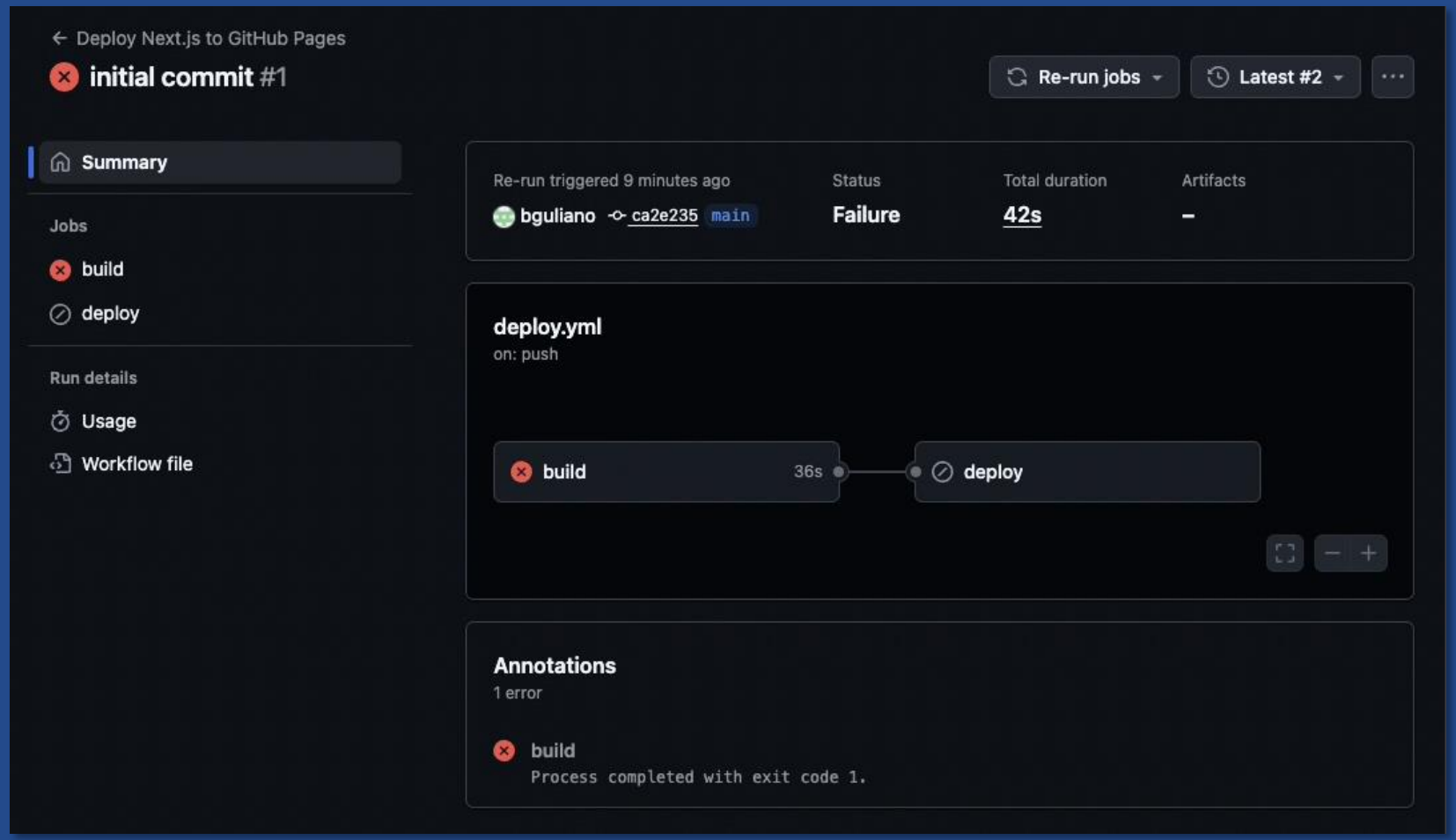
Agenda

1. Finish deploying last week's project to GitHub
2. Git and GitHub overview & KTP workflow
3. The rest of the KTP tech stack
    1. Zustand
    2. React Query
4. Upload new project to GitHub
5. Questions

Deploying to GitHub

1. Edit `next.config.js`
2. Create GitHub Actions workflow
3. Change Actions > General to Read/Write
4. Set Pages > Build and deployment > Source to GitHub Actions

Why did it fail?



Why did it fail?

```
import React from "react";

type Params = {
    projectName: string;
}

const ProjectNamePage = ({ params }: { params: Params }) => {
    return (
        <p>This page features project &#34;{params.projectName}&#34;</p>
    );
};
export default ProjectNamePage;
```

Client-side

VS

Server-side

What really is Git?

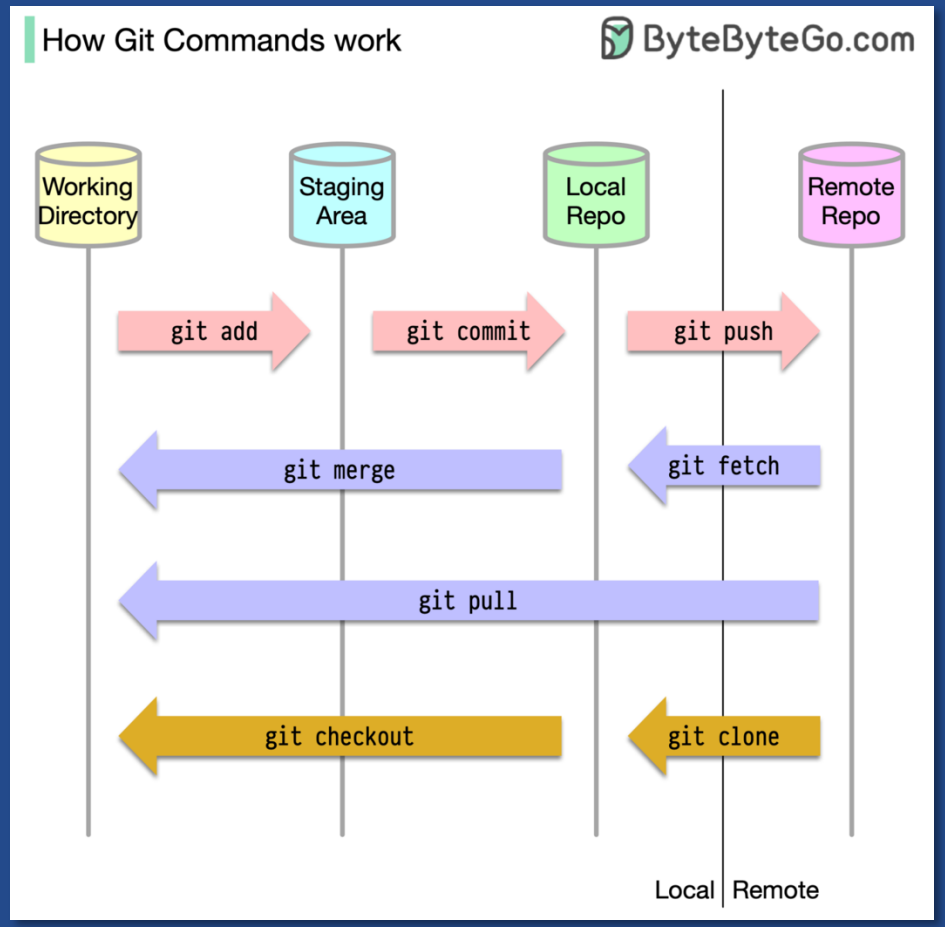
What really is Git?

- Version control system (VCS)
- Tracks changes over time
- Enables collaboration

- Avoids messy file versions
- Multiple people can work safely
- Rollback possible
- Industry standard

What really is Git?

- 4 stages
- Local vs. remote
- Tracked vs. untracked
- What WebStorm takes care of automatically



KTP’s setup

- Each project will have a repo
- Changes added via PRs
  - Each feature/change should be within a branch
  - PRs will need to be approved by at least 2 execs to make it into the main branch
- Execs = reviewers/maintainers

KTP’s setup

*We will go over how this looks in the next couple of weeks after the projects have started developing code

KTP Tech Stack

Tool	Purpose
Next.js	Core framework
TypeScript	Safer coding practices (catches errors before they occur with typing)
Tailwind CSS	Styling
Zustand	Easy and minimal global state management
React Query	Server state management and better fetching
GitHub Pages	Easy and free deployment
Figma	Design, wireframing, prototyping

Zustand

KTP Tech Stack

React Query

Zustand

- Lightweight
- Minimal boilerplate
  - Versus, i.e. useState
- Easier than Redux
  - ~~Actions~~
  - ~~Reducers~~
  - ~~Types~~
  - ~~Provider~~

React Query

- Library for **server state**
- Simplifies API calls
- Handles loading, error, success easily

Benefits

- Currently, you need to
  - Use `useEffect` and `fetch`
  - Handle loading and error
  - Write caching logic for repetitive API calls
  - Messy + boilerplate intense
- React Query, however,
  - Fetches data automatically when needed
  - Caches results
  - Manages loading/error states

Deploying to GitHub

1. Edit `next.config.js`
2. Create GitHub Actions workflow
3. Change Actions > General to Read/Write
4. Set Pages > Build and deployment > Source to GitHub Actions

Questions?