

# CI/CD and OpenAPI

Cloud Computing and SaaS



# Announcements

- Autograder Released on Gradescope
  - Tests are mostly the same as the local tests
  - Takes a while, so make sure you pass the local tests first!
- Friends Service -- Releasing tonight, latest tomorrow morning
- Frontend -- ran into unforeseen issues, trying to get it out by end of the week
- Project Deployment and AWS Neptune Deployment Instructions Released
- Grades - If we haven't reached out to you, you are fine!
-

# Last Time

- Thanksgiving Break!
- Chat with a Microsoft Recruiter
- Functions as a Service

# Today

- Extra Challenges for BearChat
- CI/CD
  - Goal, Problems Solved
  - Main Practices
  - Processes
  - Tools
- OpenAPI
  - Main Use Cases

# Today

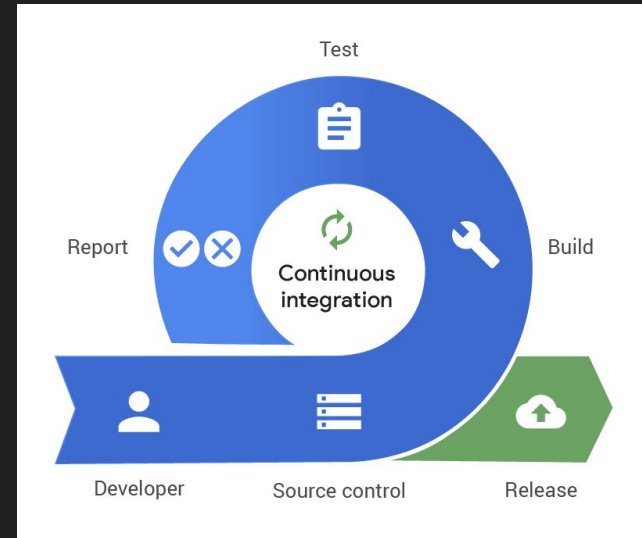
- Extra Challenges for BearChat
- CI/CD
  - Goal, Problems Solved
  - Main Practices
  - Processes
  - Tools
- OpenAPI
  - Main Use Cases

# BearChat Challenges

- Frontend
  - Implement verification and password reset endpoints
- Backend + Frontend
  - Implement Support For Images in Posts
    - AWS S3
- Backend + ML
  - Sentiment Analysis for Posts
  - Monitoring - Check Posts for inappropriate content

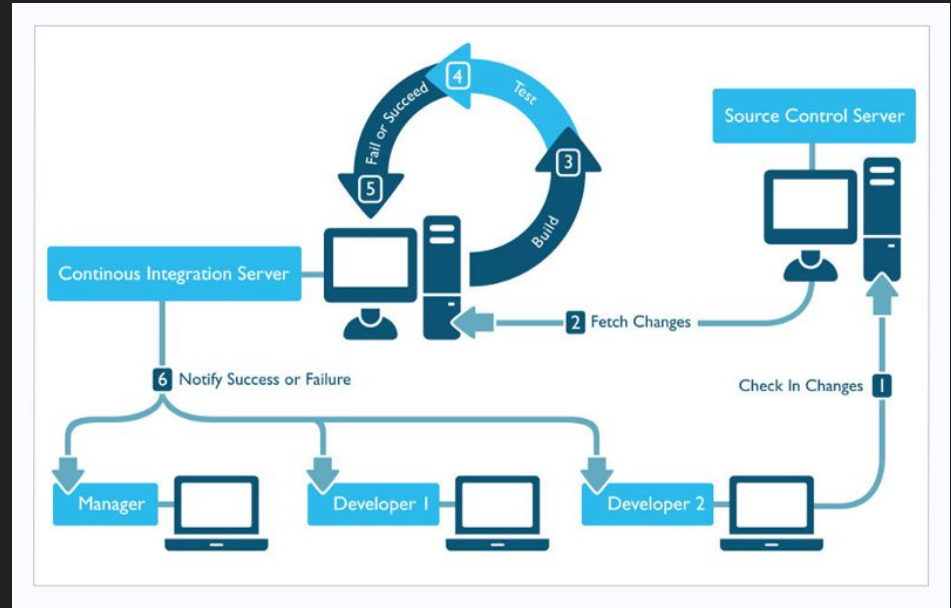
# Continuous Integration

- Goal
  - Build, Package, Test
  - Maintain Consistency
  - Encourage Better Collaboration
- Main practices
  - Develop code in small increments
  - Push to version control often



# Continuous Integration

- Process
  - Write Code
  - Push to Version Control
  - Automated Build
  - Automated Testing
  - Hand off to Continuous Delivery





# Continuous Integration

- Popular Tools
  - Jenkins
  - Travis
  - GitLab CI
  - AWS: CodeBuild + CodePipeline

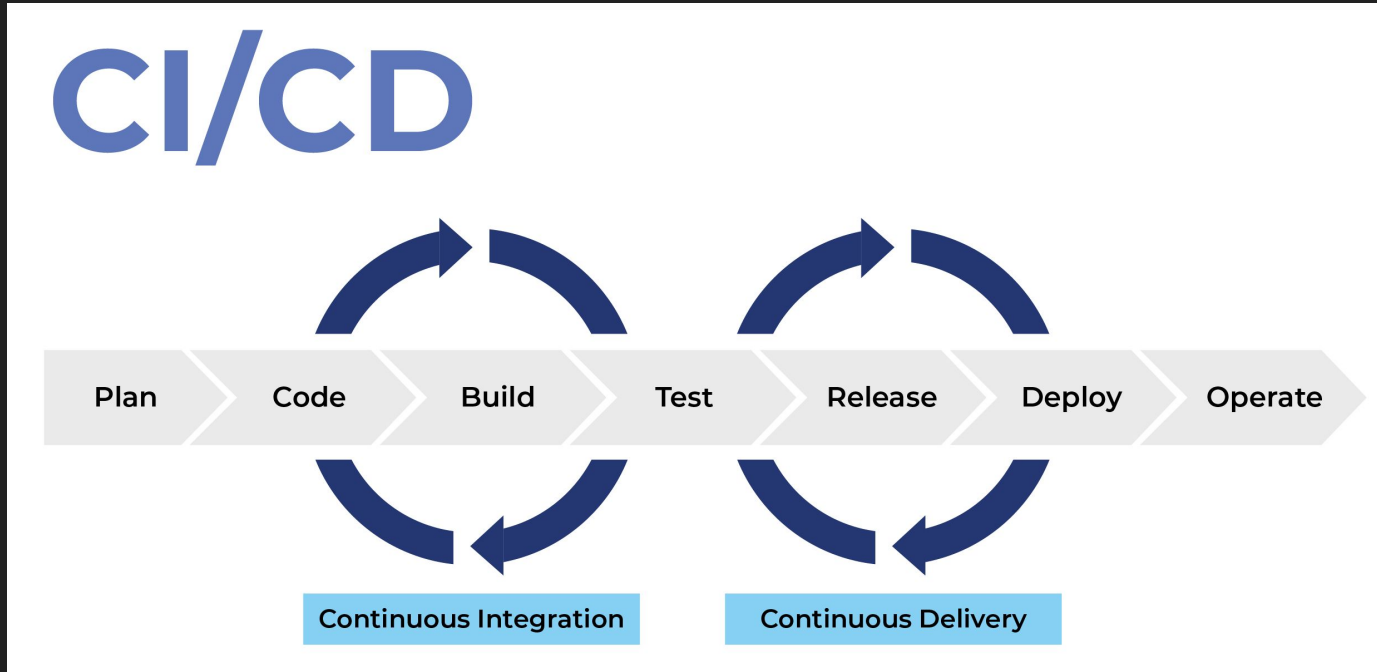


# Continuous Deployment

- Goal
  - Pick up where continuous integration left off
  - Automated Delivery to Infrastructure Environments
    - Testing, Production, Release
- Main Practice: Keep codebase deployable at any point
  - Make sure it passes all build tests

# Continuous Deployment

- Process (with continuous delivery)

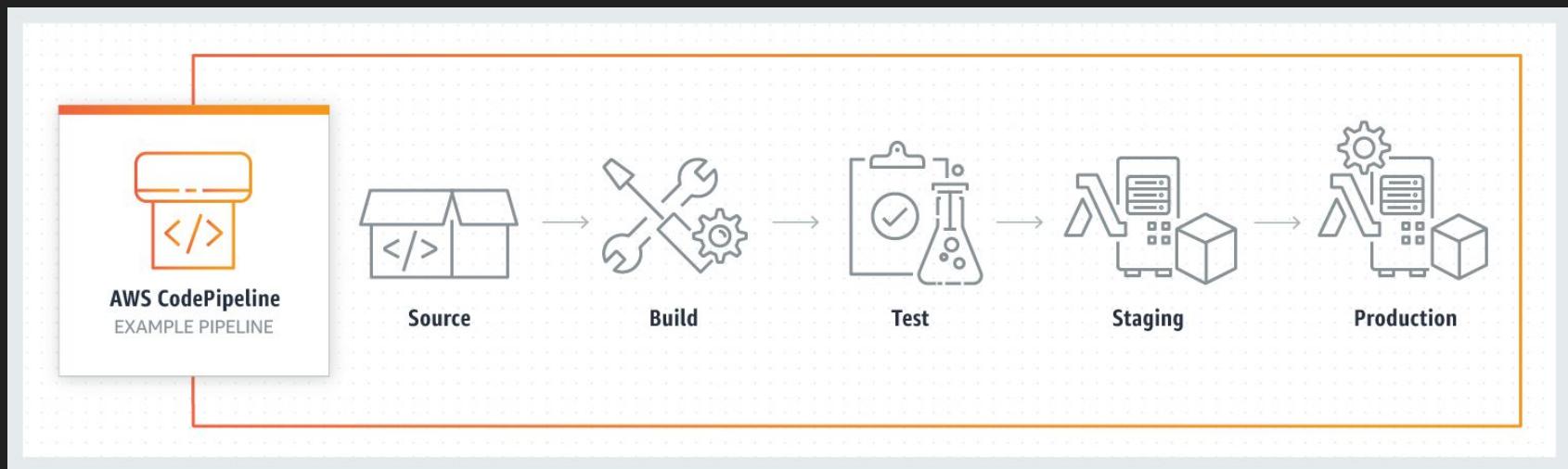


# Continuous Deployment

- Popular Tools
  - Jenkins
  - CircleCI
  - TeamCity
  - AWS: CodeBuild + CodePipeline
- CI/CD tools are usually packaged together,
  - Often as a single product

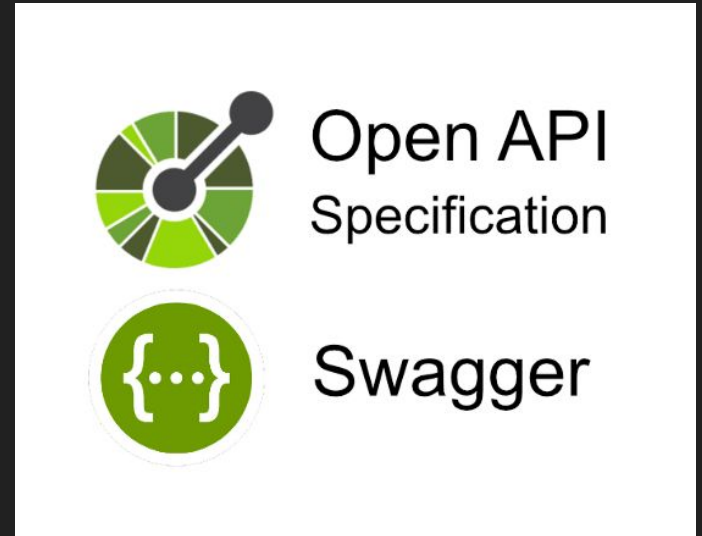


# AWS CI/CD



# OpenAPI

- A set of standards/conventions
  - OpenAPI Specification (OAS)
- Understand services without access to code or documentation
- Often used to design API's
  - You may be asked to do this as a full-time SWE or PM
  - Spec's are sometimes called an "API Contract"



# OpenAPI - Structure

- Usually written as a single file
  - Choice of YAML or JSON
- At the top: OpenAPI Version
- Info - title/other metadata about the spec
- Servers - API server, where the requests are going
- Paths -> Routes
  - BearChat: defined at top of "api.go" files

```
1. openapi: 3.0.0
2. info:
3.   title: Sample API
4.   description: Optional multiline or single-line description in [CommonMark](http://commonma
5.   version: 0.1.9
6.
7. servers:
8.   - url: http://api.example.com/v1
9.     description: Optional server description, e.g. Main (production) server
10.  - url: http://staging-api.example.com
11.    description: Optional server description, e.g. Internal staging server for testing
12.
13. paths:
14.   /users:
15.     get:
16.       summary: Returns a list of users.
17.       description: Optional extended description in CommonMark or HTML.
18.       responses:
19.         '200': # status code
20.           description: A JSON array of user names
21.           content:
22.             application/json:
23.               schema:
24.                 type: array
25.                 items:
26.                   type: string
```

# OpenAPI - Structure

- Request Body

```
paths:
  /users:
    post:
      summary: Creates a user.
      requestBody:
        required: true
        content:
          application/json:
            schema:
              type: object
              properties:
                username:
                  type: string
      responses:
        '201':
          description: Created
```



# OpenAPI - Structure

- Responses

```
responses:  
  '200':  
    description: A user object.  
    content:  
      application/json:  
        schema:  
          type: object  
          properties:  
            id:  
              type: integer  
              format: int64  
              example: 4  
            name:  
              type: string  
              example: Jessica Smith  
  '400':  
    description: The specified user ID is invalid (not a number).  
  '404':  
    description: A user with the specified ID was not found.  
  default:  
    description: Unexpected error
```

# OpenAPI - Example

- Early Version of BearChat
  - <https://app.swaggerhub.com/apis-docs/berkeleyclouddecal/bear-chat/1.0#>

Auth	
POST	<code>/auth/signup</code> Sign up for an account
POST	<code>/auth/signin</code> Sign into your account
POST	<code>/auth/sendreset</code> Begin password reset process
POST	<code>/auth/resetpw</code> Reset your password
POST	<code>/auth/verify</code> Verify a User's Email
posts	
POST	<code>/api/posts/create</code> Create a Post
GET	<code>/api/posts/{startIndex}</code> View Posts from Newsfeed
GET	<code>/api/posts/{userID}/{startIndex}</code> See most recent Posts from a User
DELETE	<code>/api/posts/delete/{postid}</code> Delete a Post

# OpenAPI - Example

## Auth

**POST** /auth/signup Sign up for an account

Sign up for BearChat

### Parameters

No parameters

Request body

Example Value | Schema

```
{
  "email": "string",
  "password": "string",
  "username": "string"
}
```

Code Description

201 Created

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "success": "string"
}
```

401 Unauthorized

Media type

application/json

Example Value | Schema

```
{
  "error": "string"
}
```