

# AWS Visual Configuration App

Iteration 3

CS 498

# Team Members

- Joshua Kennedy
- Benjamin Furlani
- Wyatt Lawrence
- Nick Hammerstrom
- Noah Connolly
- Ben Lawson (Mentor)

## Client Information

- Trey Gourley
- Accutech Systems, Muncie, IN

## Mentor Feedback

- Mentor recommended we implement stop/start functionality for the few services that have that ability.
- Mentor's opinion was that error logging should not be as high a future priority as making the UI look nicer.
- Recommended that we make MRs and code reviews a regular thing.

# Client Feedback

- After discussing our idea of a 3-tier tree for ECS hierarchy, client said he preferred a click-and-explode thing, somewhat like a wagon wheel.
- Would like to see more color in the UI.
- Was not interested in start/stop functionality, for security reasons. This would get into permissions, and it shouldn't be easy for the user to accidentally stop a service.
- Wants Route 53 for the next iteration, explained how he would like it visualized, as an icon (representing a Route 53 instance), with a number on it (representing that instance's number of zones).

# Future - Fourth Iteration Features

- ECS “wagon wheel” expanded view
- Route 53 (new service)
- More refinement for the GUI - color, AWS icons and logos perhaps
- Error logging
- Stop/start functionality for applicable AWS services
- Make Dev Mode option environment variable-based (rather than clicking the “o” in “Login” three times)

# Teamwork - Work Distribution

- Joshua Kennedy: refactor EC2, RDS, LB handlers, create mock clients for each, and improve testing for each.
- Wyatt Lawrence: refactor ECS, improve ECS testing, begin developing ECS “wagon wheel” hierarchy on separate branch
- Nick Hammerstrom: mock ECS client (clusters and services), investigate project compatibility with Docker
- Noah Connolly: S3 handler
- Benjamin Furlani: Auto-refresh feature

# Shit Happens

- Before turning in the software (due February the 8th at 11:59 PM), we ran our software a last time, and all its tests, and discovered that all of our mock AWS clients were throwing errors. This caused many tests to fail, and for mock items to not show up in the dashboard.
- This is because AWS released an update for the .NET SDK earlier that day (<https://github.com/aws/aws-sdk-net>) , which added many new methods to services (especially EC2), methods which our mock clients were not implementing.



# Random: 1920s Futurism Urban Planning



# 1920s Futurism Urban Planning

