

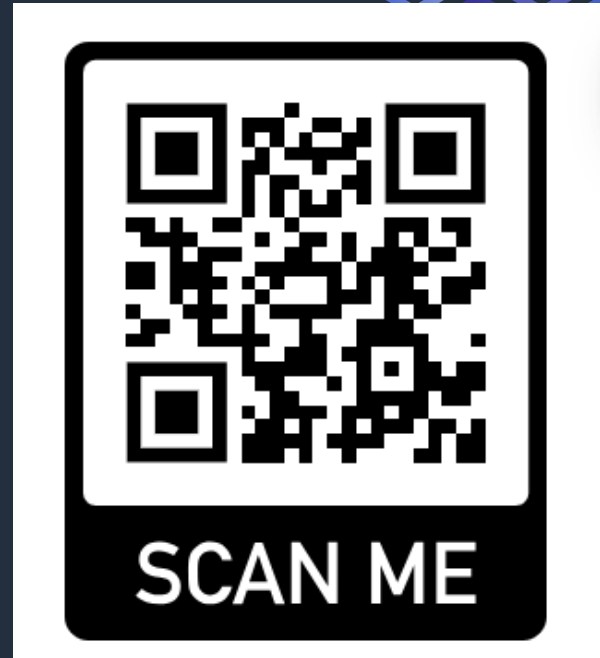


# COMMUNITY DAY

AOTEAROA

## Gamifying Unicorns and Rainbows using AWS EKS

STEFAN EVANS | DEVOPS CYBERCX | 6<sup>th</sup> September 2023



## Welcome to the beta game launch!

Please get out your phone and use your camera to scan the QR code above and join the game



# COMMUNITY DAY

AOTEAROA

STEFAN EVANS  
MANAGING CONSULTANT



CyberCX



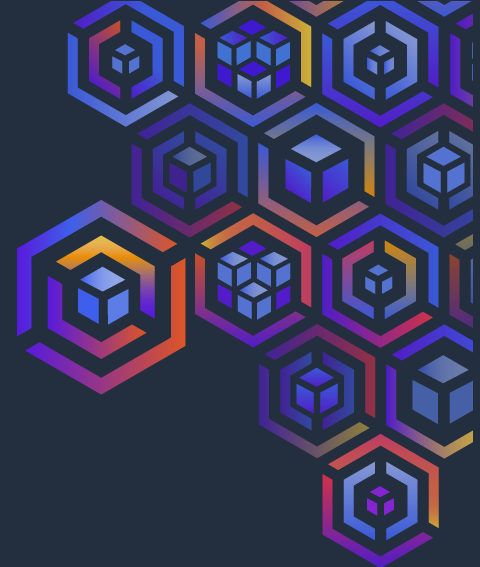
## Topics I'll cover:

- Why did I decide to build a game?
  - Family
  - Fun
  - Further learning
- What in this session for you?
  - Unicorns and Rainbows
  - Javascript and WebSockets
  - AWS Elastic Kubernetes Service (EKS)
  - AWS Code Pipeline



# COMMUNITY DAY

AOTEAROA





# COMMUNITY DAY

AOTEAROA

STEFAN EVANS  
MANAGING CONSULTANT



CyberCX

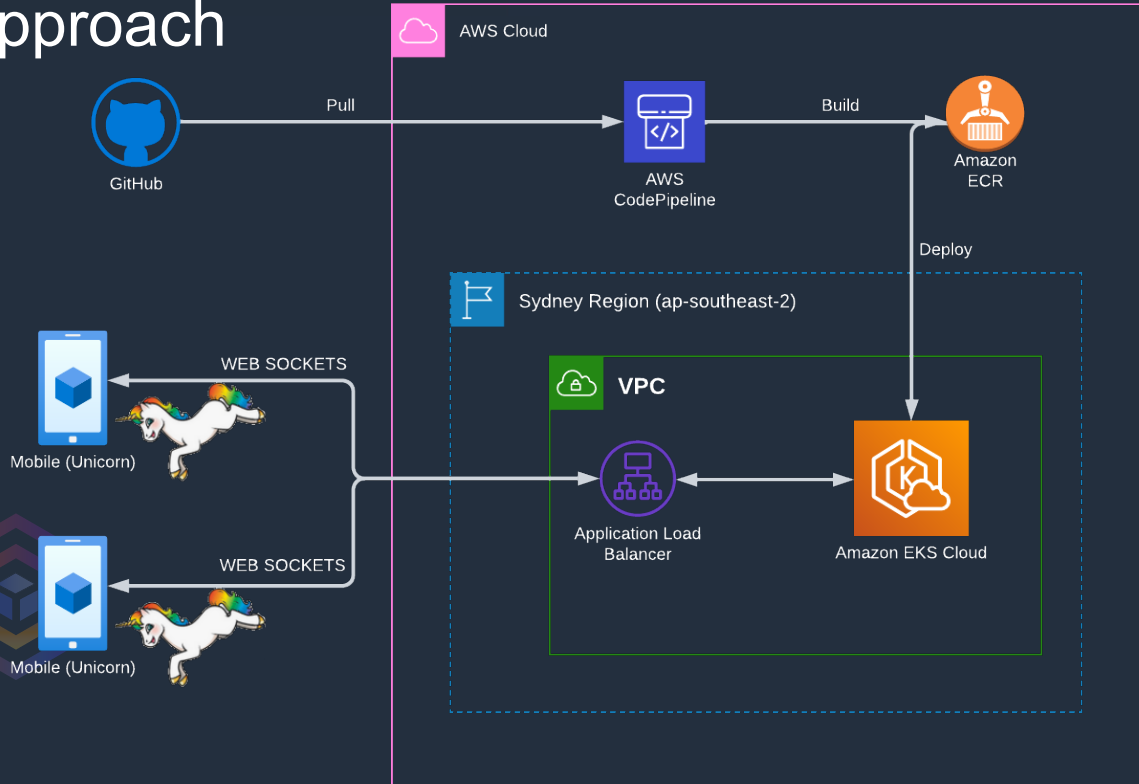


## Topics I'll cover:

- Why did I decide to build a game?
  - Family
  - Fun
  - Further learning
- What in this session for you?
  - Unicorns and Rainbows
  - Javascript and WebSockets
  - AWS Elastic Kubernetes Service (EKS)
  - AWS Code Pipeline

aws  
**COMMUNITY DAY**  
AOTEAROA

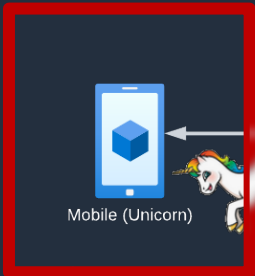
# My Approach





# COMMUNITY DAY

AOTEAROA



Phaser is a 2D game framework used for making HTML5 games for desktop and mobile.

**Desktop and Mobile HTML5 game framework**

A fast, free and fun open source framework for Canvas and WebGL powered browser games.

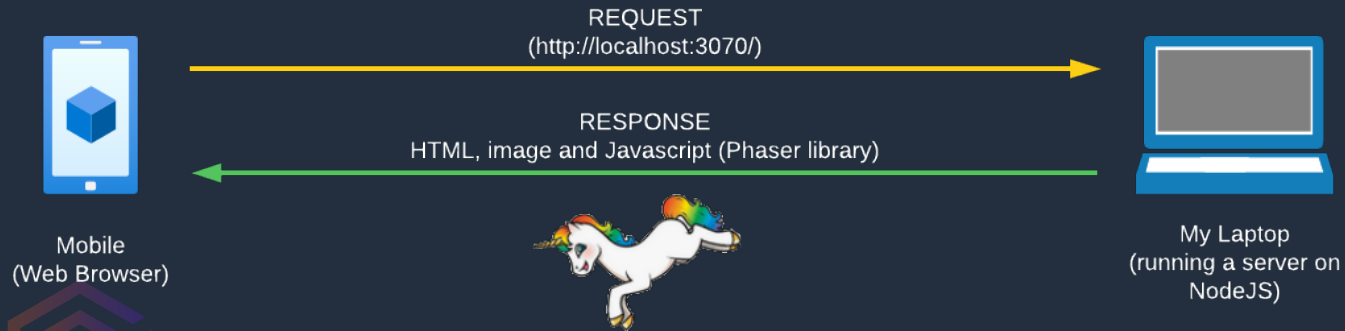
**DOWNLOAD & GET STARTED**  
Download or Fork via Github

3,600

HTML5

The banner features a colorful illustration of a purple alien character, a green alien character, and a rocket ship on a planet. The word 'PHASER' is written in large, 3D block letters. The HTML5 logo is in the bottom left corner.

# What is the Phaser library?







# COMMUNITY DAY Phaser Deep Dive

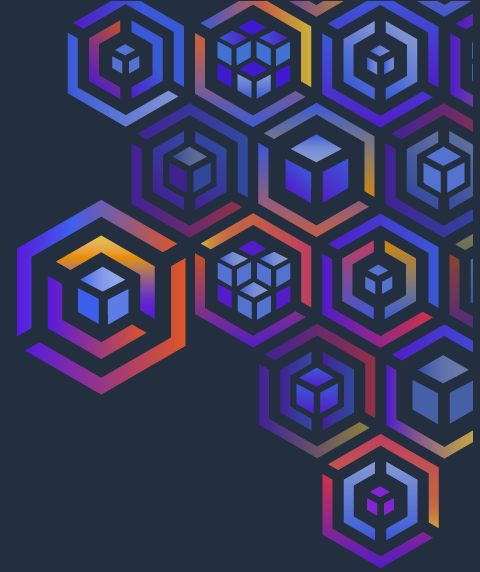
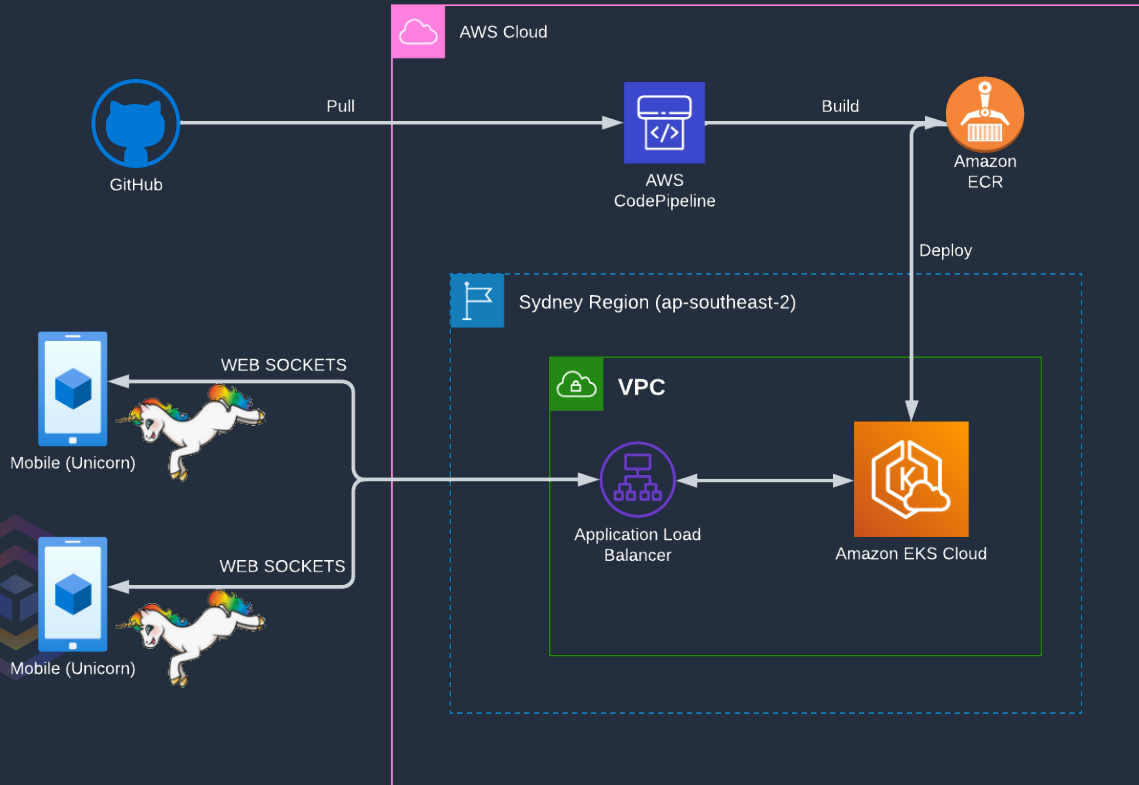
AOTEAROA

The screenshot shows a code editor interface with the following components:

- EXPLORER:** A file tree on the left showing a project structure under "1. NEW GAME". The tree includes:
  - node\_modules
  - public
    - assets
    - js
      - game.js
      - index.html (selected)
  - .gitignore
  - package-lock.json
  - package.json
  - README.md
  - server.js
- EDITOR:** The main workspace displays the content of `index.html`. The code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="utf-8">
6 </head>
7
8 <body>
9   <script src="//cdn.jsdelivr.net/npm/phaser@3.60.0/dist/phaser.min.js"></script>
10  <script src="js/game.js"></script>
11 </body>
12
13 </html>
14
```
- TERMINAL:** A terminal window at the bottom is open, showing a prompt `>` and a cursor.
- Navigation:** The bottom of the editor has tabs for `TERMINAL`, `CODEWHISPERER`, `REFERENCE LOG`, `GITLENS`, `OUTPUT`, `DEBUG CONSOLE`, and `COMMENTS`.

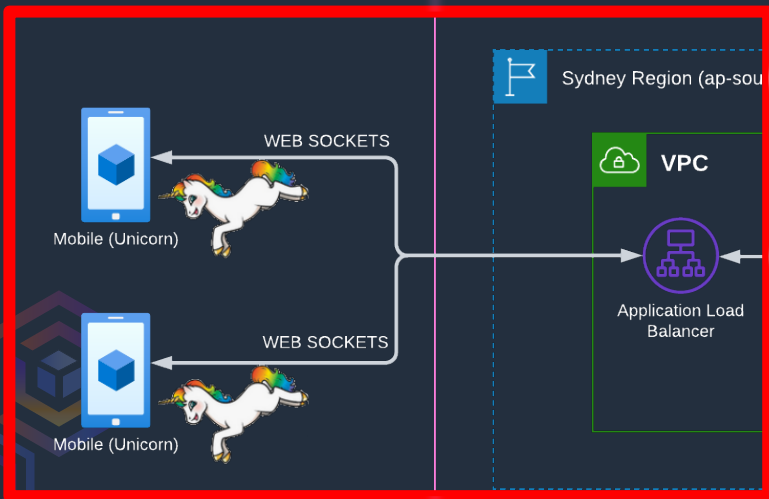
aws  
**COMMUNITY DAY**  
AOTEAROA





# COMMUNITY DAY

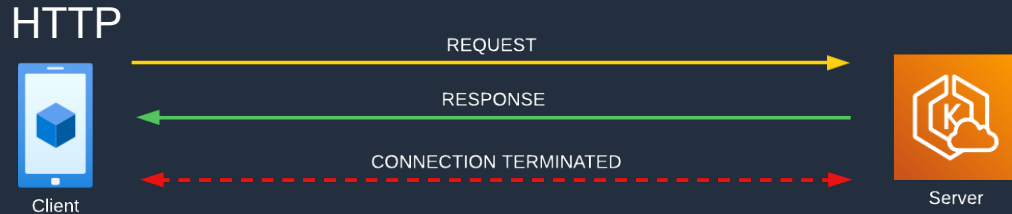
AOTEAROA



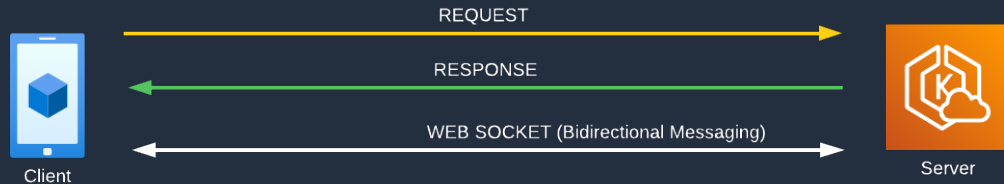
**WebSockets** makes it possible to open a two-way interactive communication session between the user's browser and a server.

You can send messages to a server and receive event-driven responses without having to poll the server for a reply.

# What are WebSockets?



## Web Sockets





# COMMUNITY DAY

## WebSocket Deep Dive

AOTEAROA

The screenshot displays a development environment with the following components:

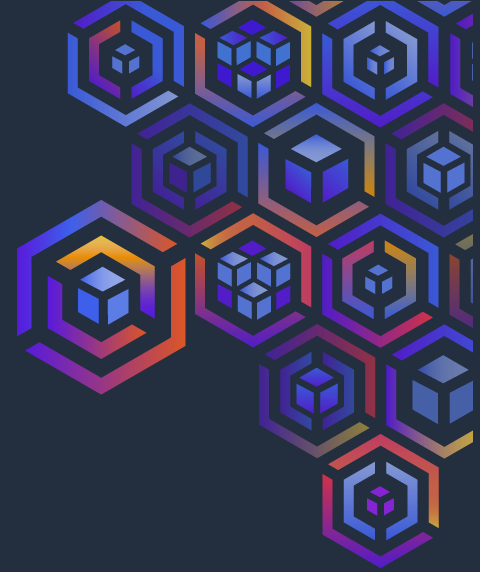
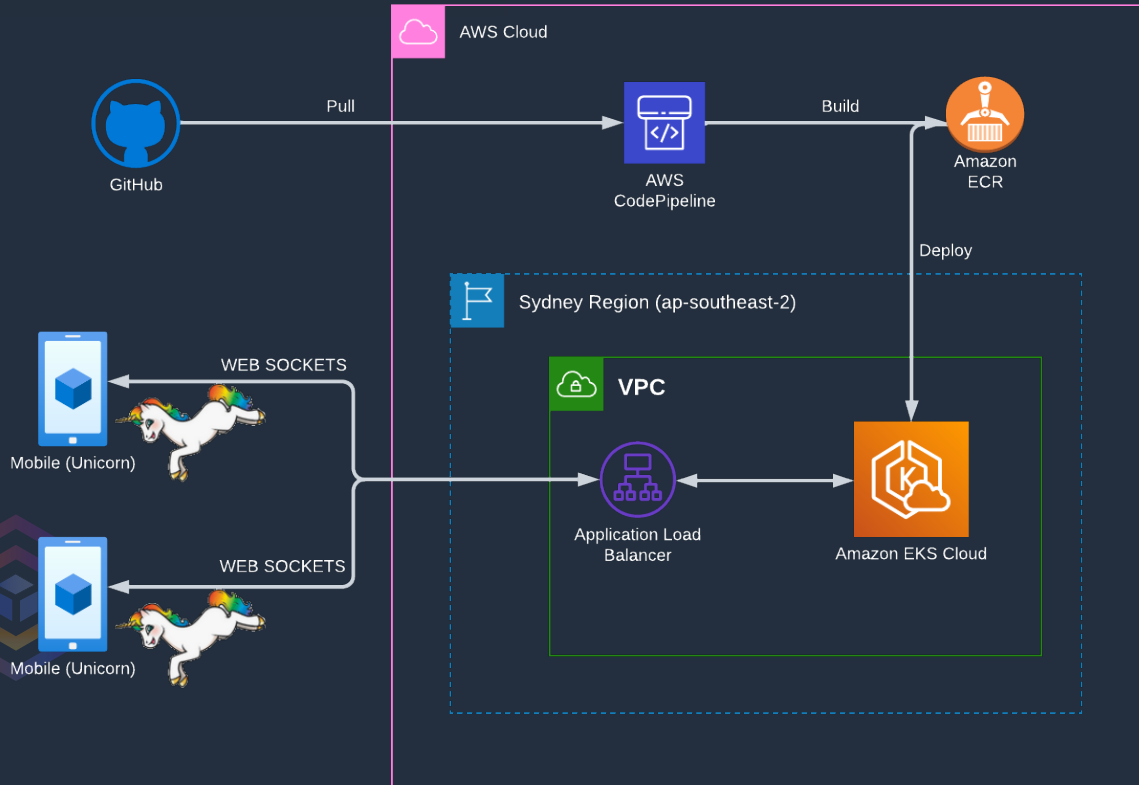
- EXPLORER:** Shows a project structure for '1. NEW GAME' with folders for 'node\_modules', 'public', 'assets', and 'js'. The 'js' folder contains 'game.js' and 'index.html'. Other files include 'unicorn.png', '.gitignore', 'package-lock.json', 'package.json', and 'README.md'.
- Code Editor:** Displays the content of 'server.js' with the following code:

```
1 var express = require('express');
2 var app = express();
3 var server = require('http').Server(app);
4 // var io = require('socket.io').listen(server);
5
6 app.use(express.static(__dirname + '/public'));
7
8 app.get('/', function (req, res) {
9   res.sendFile(__dirname + '/index.html');
10 });
11
12 server.listen(3070, function () {
13   console.log('Game started and is listening on ' + server
14 });
15
16 // io.on('connection', function (socket) {
17 //   console.log('websocket connection created with id: '
18
```
- TERMINAL:** Shows the execution of the server:

```
> phaser_game@1.0.0 start
> node server.js

Game started and is listening on 3070
█
```
- Browser:** A browser window at 'localhost:3070' displays a blue background with a cartoon duck character.

aws  
**COMMUNITY DAY**  
AOTEAROA

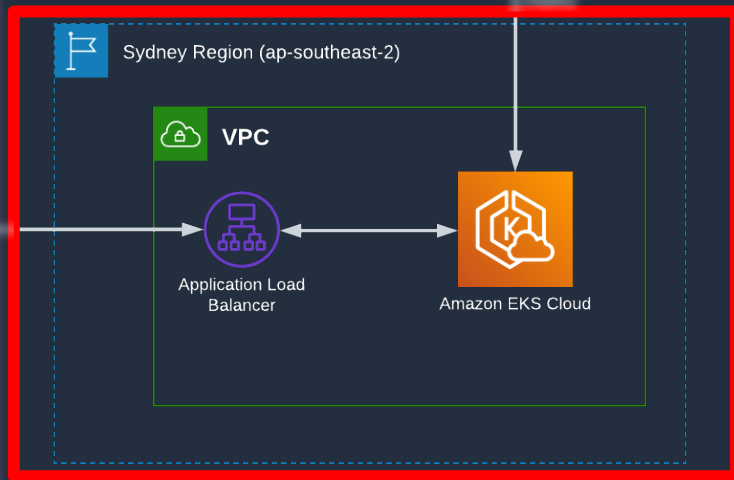




# COMMUNITY DAY

AOTEAROA

Amazon Elastic Kubernetes Service (Amazon EKS) is a managed Kubernetes service to run Kubernetes in the AWS



## So why AWS EKS (Elastic Kubernetes Service)?

- Containers
- AWS Managed Kubernetes
- Being used more by our customers





# COMMUNITY DAY

AOTEAROA

## AWS EKS Deep Dive

File Edit Selection View Go Run Terminal Help

www.BANDICAM.com

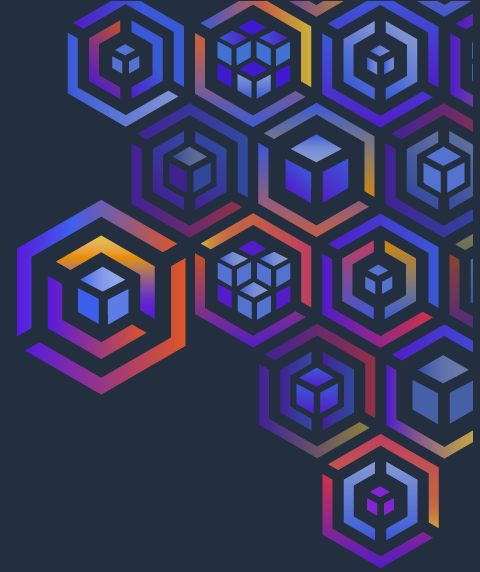
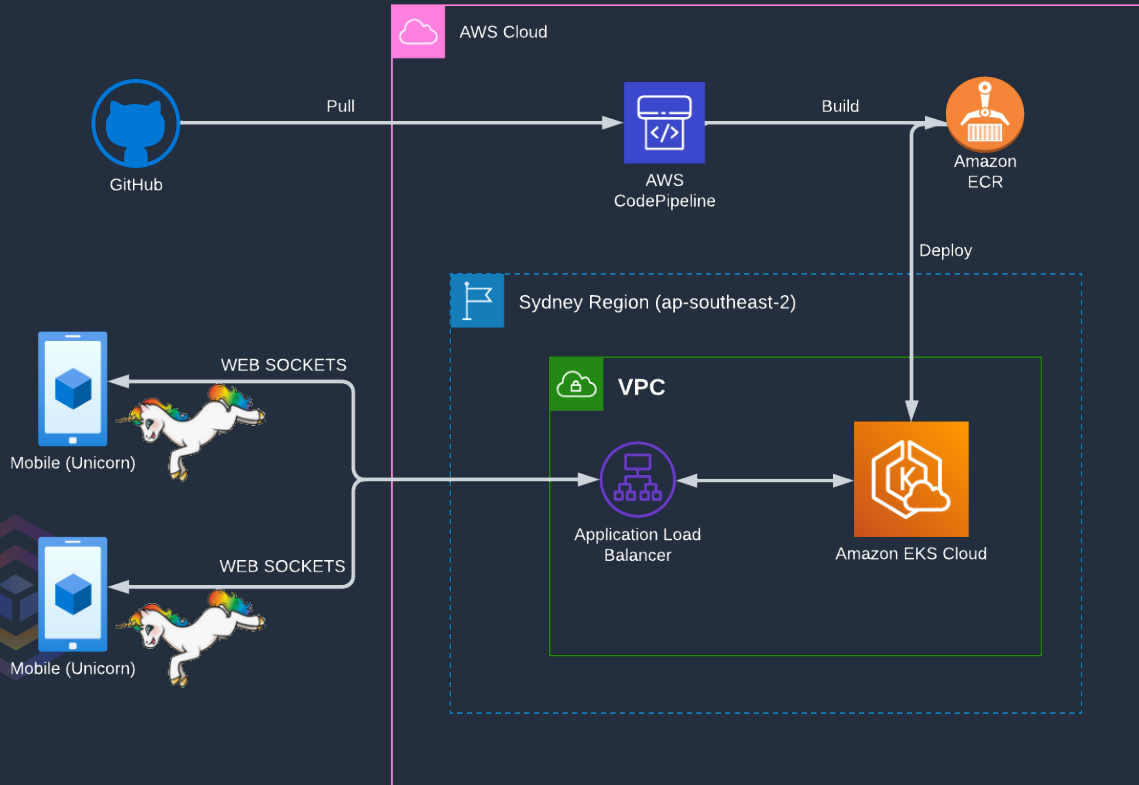
EXPLORER

- PHASERSOCKETIOGAME
  - buildspec
  - kubernetes
  - node\_modules
  - public
  - terraform
  - terraform-cicd-pipeline
  - .gitignore
  - Dockerfile
  - package-lock.json
  - package.json
  - README.md
  - server.js

TERMINAL

```
PS C:\Users\StefanEvans\Development\ambassador\unicorns\phasersocketio\terraform> terraform apply
```

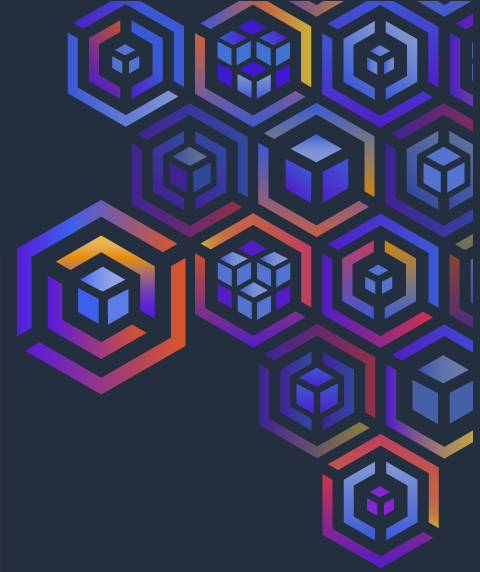
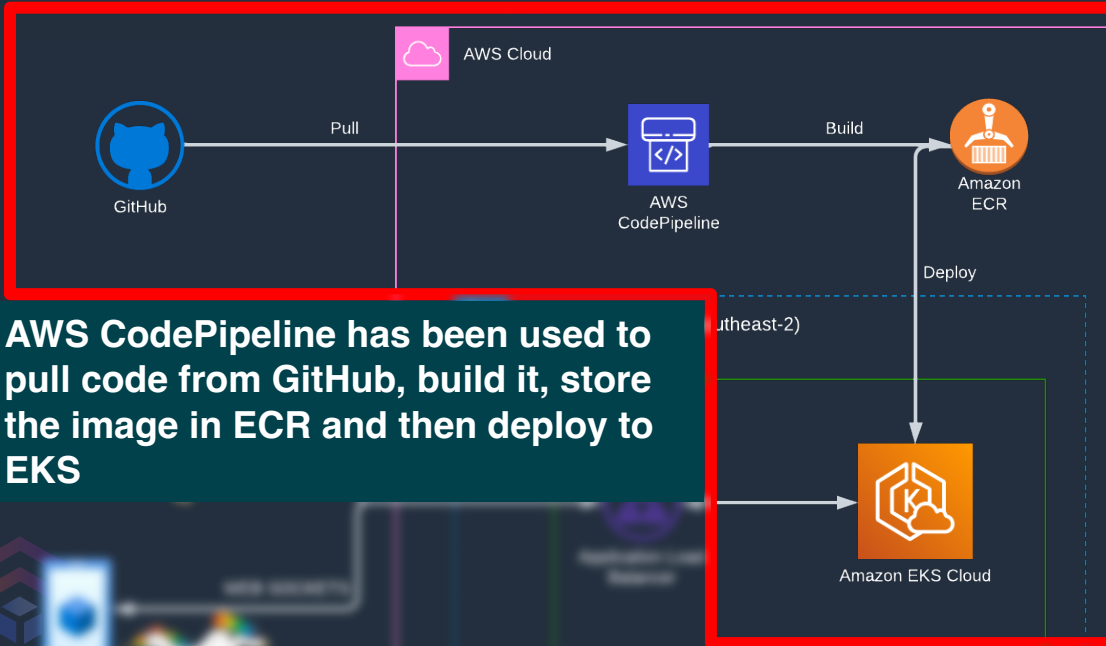
aws  
**COMMUNITY DAY**  
AOTEAROA





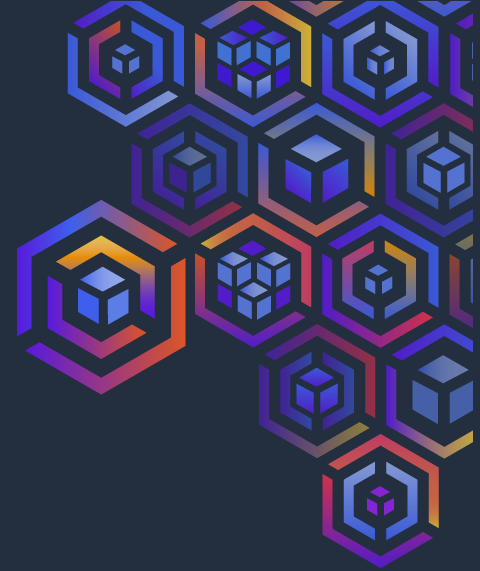
# COMMUNITY DAY

AOTEAROA



## So why AWS CodePipeline?

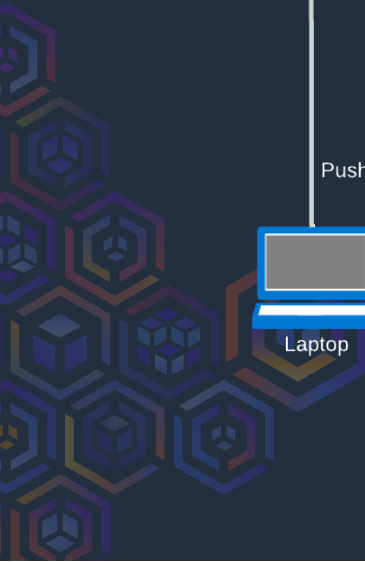
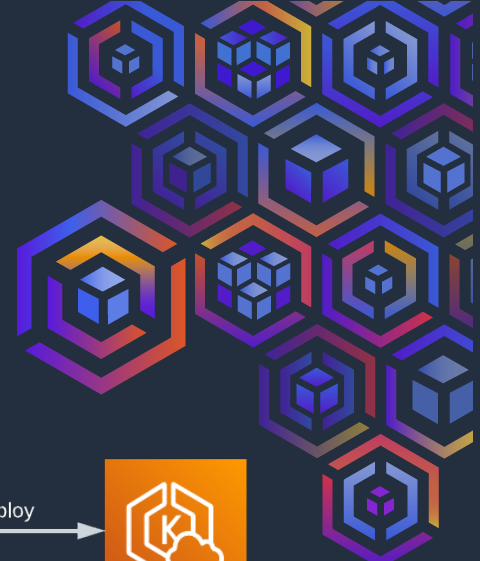
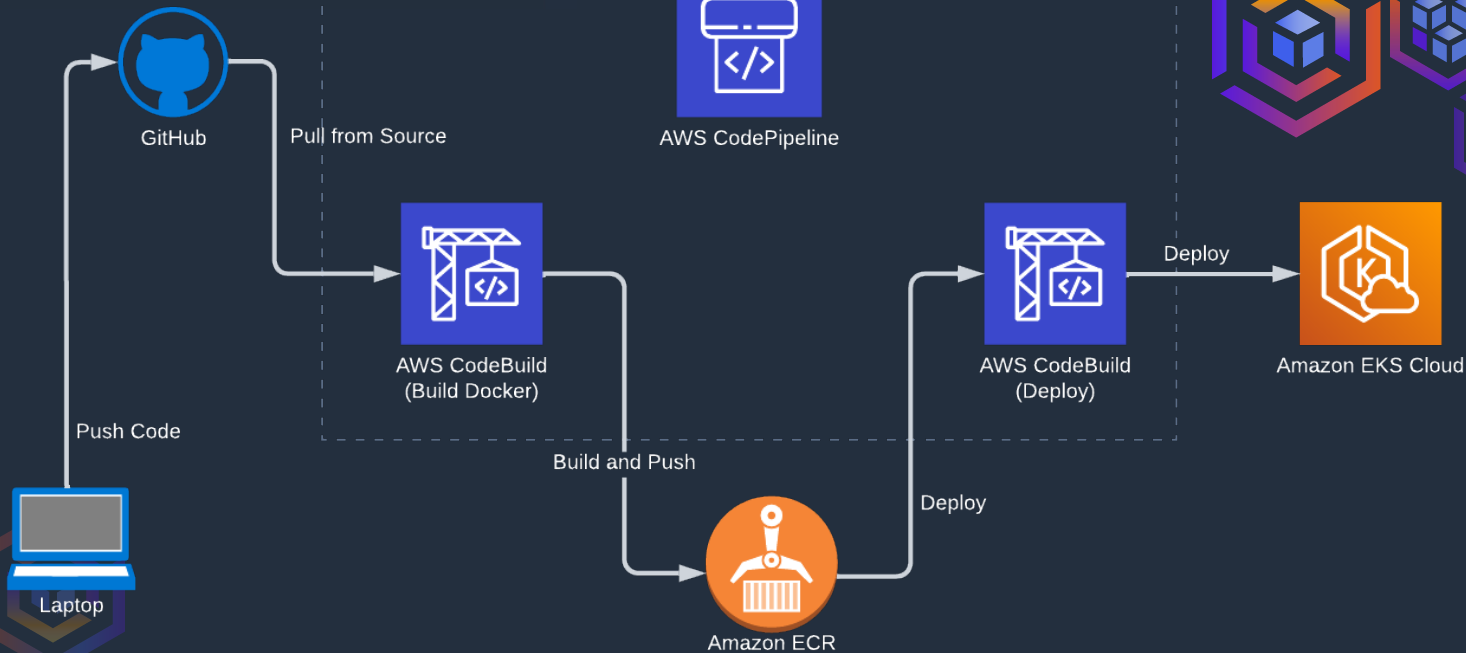
- Automated deployment
- Works well with services like ECR and IAM
- AWS service for CI/CD





# COMMUNITY DAY

AOTEAROA





# COMMUNITY DAY

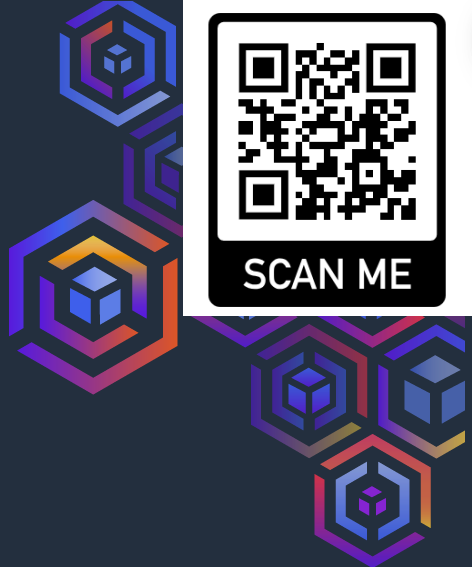
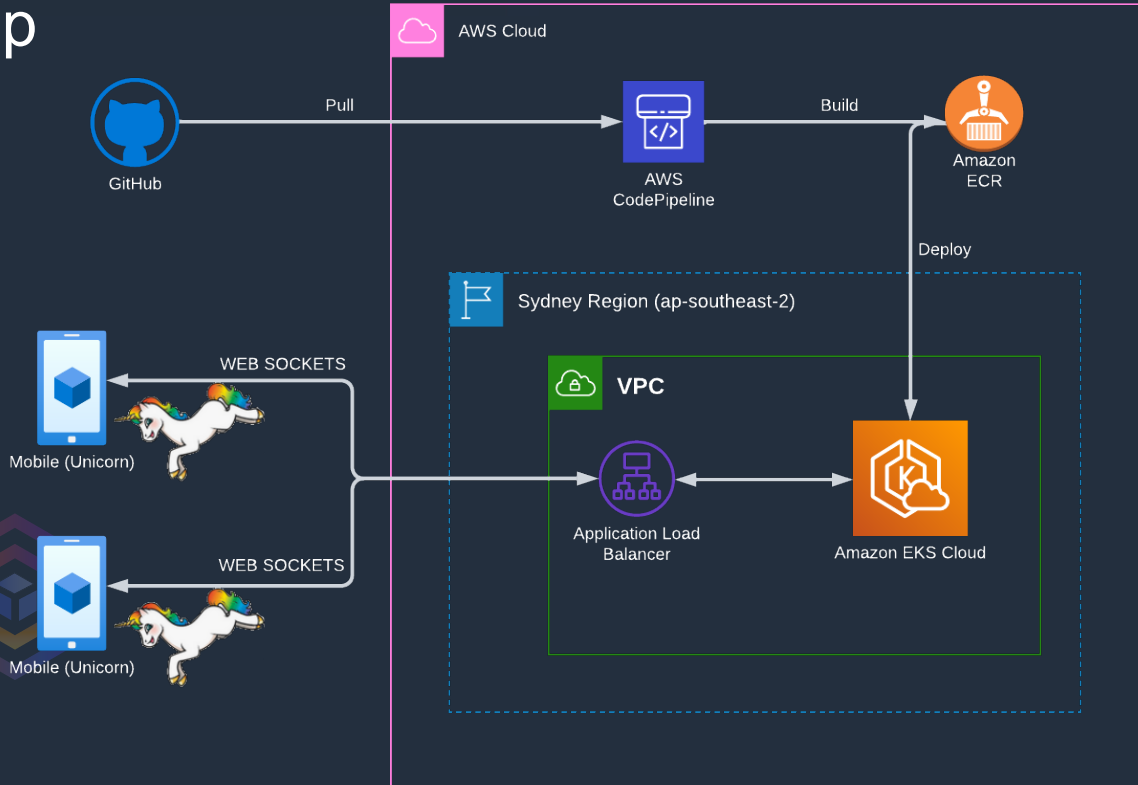
AOTEAROA

## CodePipeline Deep Dive

```
public > js > game.js > UnicornGame > version
1
2 class UnicornGame extends Phaser.Scene
3 {
4   GAME_WIDTH;
5   GAME_HEIGHT;
6
7   version = '0.7';
8
9   backgroundScene;
10  parent;
11  sizer;
12  player;
13  playerId;
14  playerName;
15
16  otherPlayers;
17
18  inputText;
19  inputName = '';
20
21  joystickState = '';
22
23  preload() {
24    this.load.image('player', 'assets/unicorn.png');
25    this.load.image('otherPlayer', 'assets/unicorn.png');
26    this.load.image('goal', 'assets/rainbow.png');
27
28    var url;
29    url = 'https://raw.githubusercontent.com/rexrainbow/phaser3-rex-notes/master/dist/rexvirtualjoys
30    this.load.plugin('rexvirtualjoystickplugin', url, true);
31
32    this.load.plugin('rexinputtextplugin', 'https://raw.githubusercontent.com/rexrainbow/phaser3-re
33  }
34
35  create() {
```

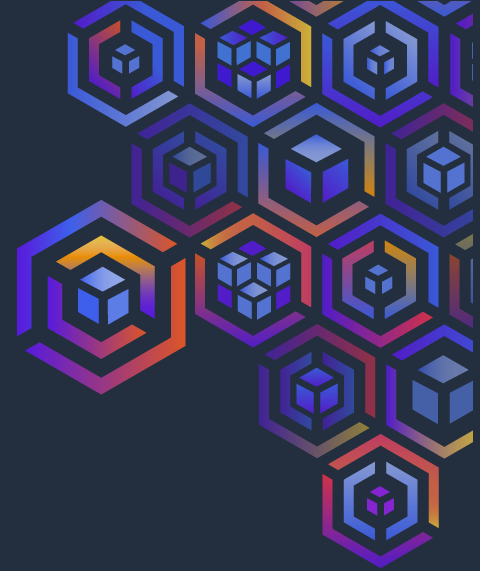
aws  
**COMMUNITY DAY**  
AOTEAROA

# Recap



What I learnt from all this.

- Writing games is a craft!
- Scalability is a challenge
- Keep learning by making mistakes



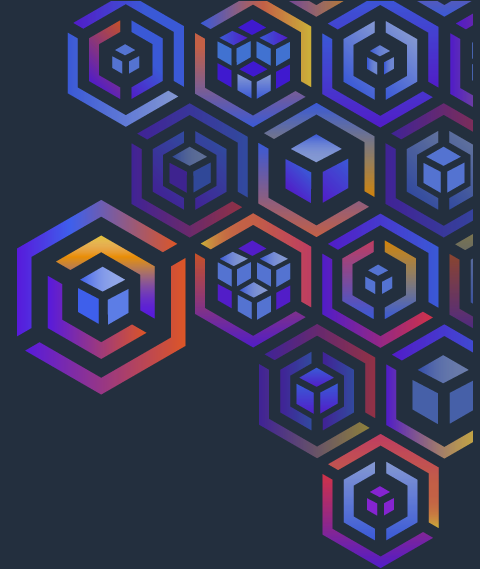




# COMMUNITY DAY

AOTEAROA

Thank you and Questions?





# COMMUNITY DAY

AOTEAROA