



Messaging Backbone Runbook

# Introduction

This document is intended to be your guide to supporting your work and your team while you are employed at Unicorn.Rentals. The information in this document is *intended* to be factual. All the information you need to do your job well may not be in this document but it is a good start.

# Price List

Here at Unicorn Rentals we know that it takes money to make money. Outlined below are the various costs you will incur for running infrastructure in your account. So make sure you only provision what you need!

Every 5 minutes, you will be charged the following:

* 30 Points for every EC2 Instance
* 10 Points for every provisioned DynamoDB WCU
* 10 Points for every provisioned DynamoDB RCU
* 30 Points for every ElastiCache Node

# Base Infrastructure Topology

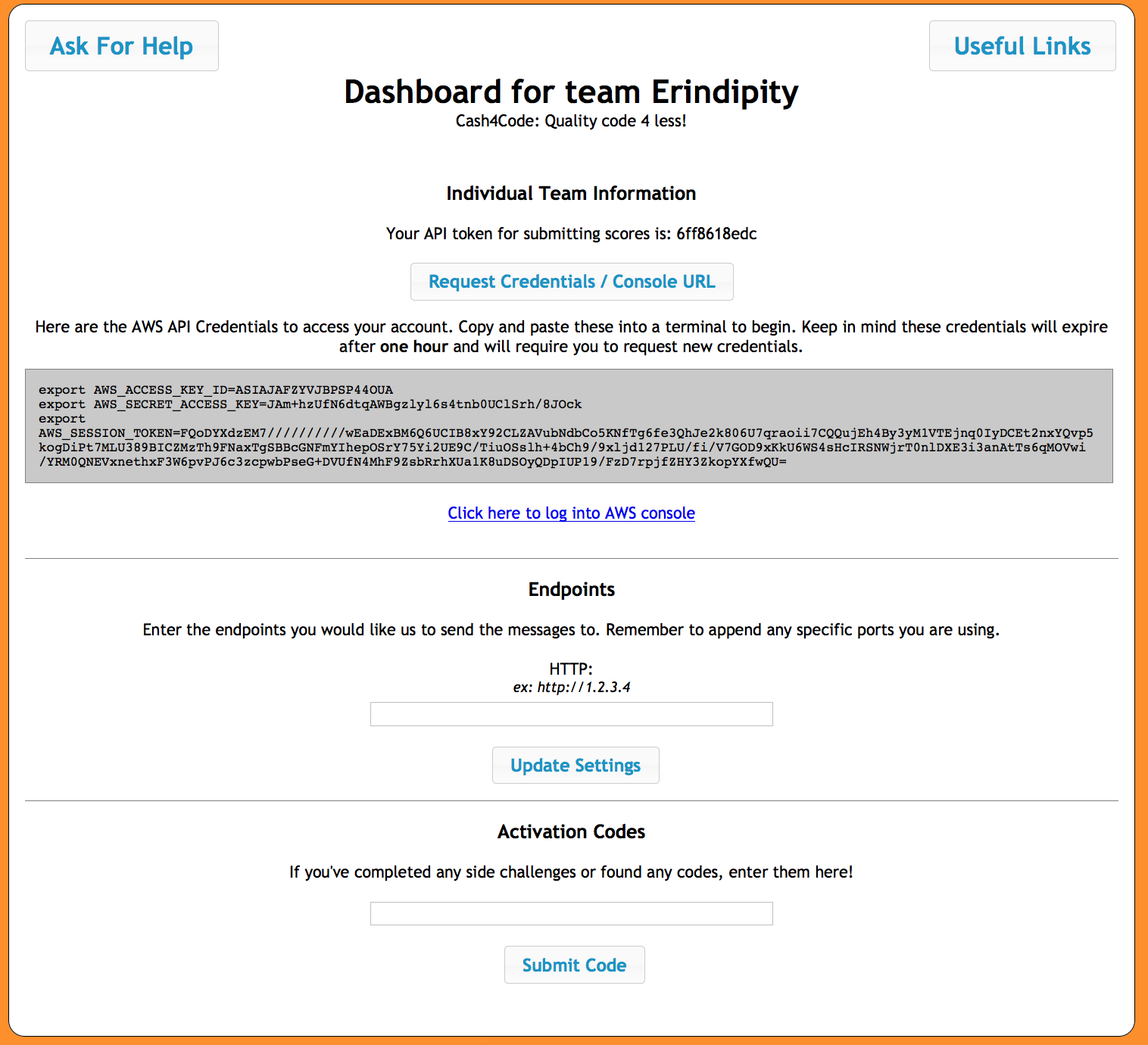
**Availability Zone – us-west-2a**

Some of the previous employees left some infrastructure running in the account. We found the drawing below on a napkin labeled “Maybe?” so dive into the account and verify what exactly is running. As this is Unicorn Rentals….it most likely will have problems.



# Player Dashboard

The dashboard can be accessed by going to https://dashboard.cash4code.net. It will prompt you to enter your Team Identifier. This team identifier is your key to the game and will be given to you by Unicorn.Rentals management. Do not share this with anyone else! The second prompt is to name your team. Keep in mind that this team name cannot be changed once set and will represent your team on the scoreboard. The scoreboard team at Unicorn.Rentals did not have the motivation to make the scoreboard private so keep in mind it is a public site. Any inappropriate team names will be renamed by the Unicorn.Rentals board of directors and we may tell your parents that you thought it was a good idea to put bad words on the Internet. Shame on you.



21

1

31

431

531

6

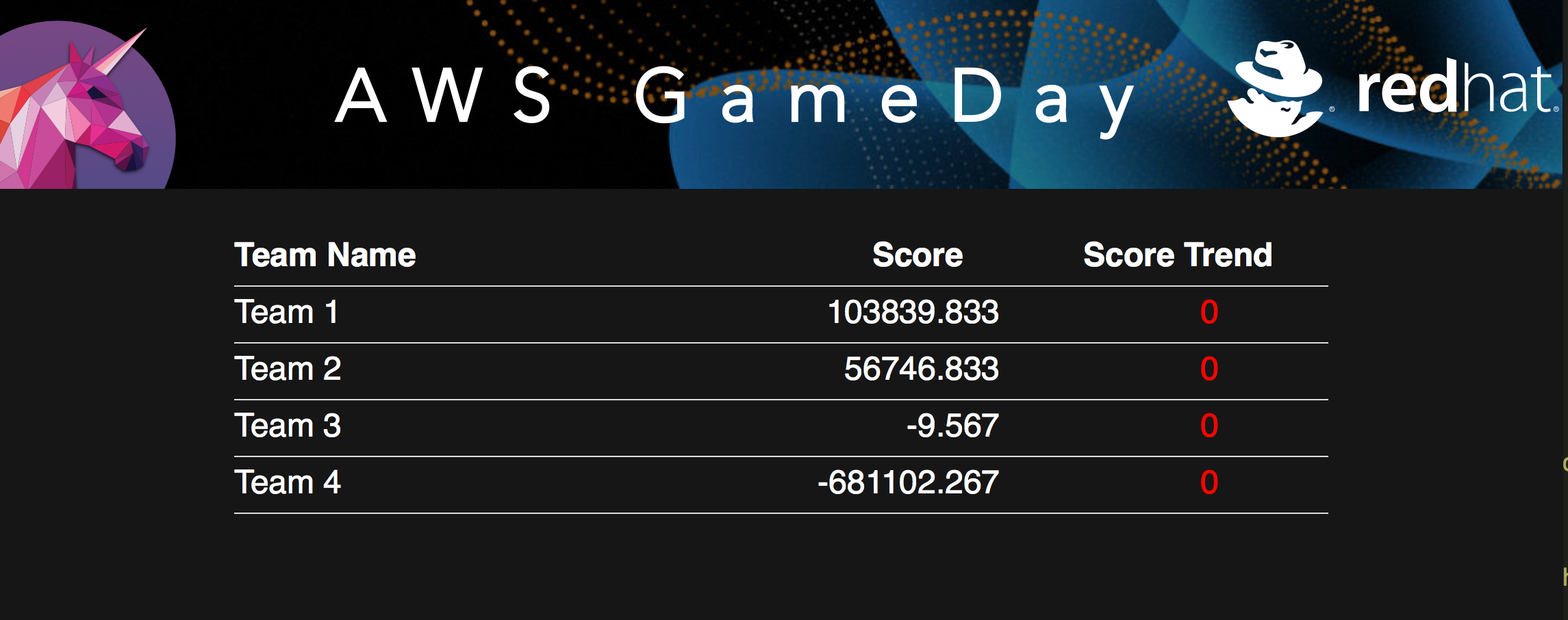
The dashboard has a few key components that you will interact with throughout the game.

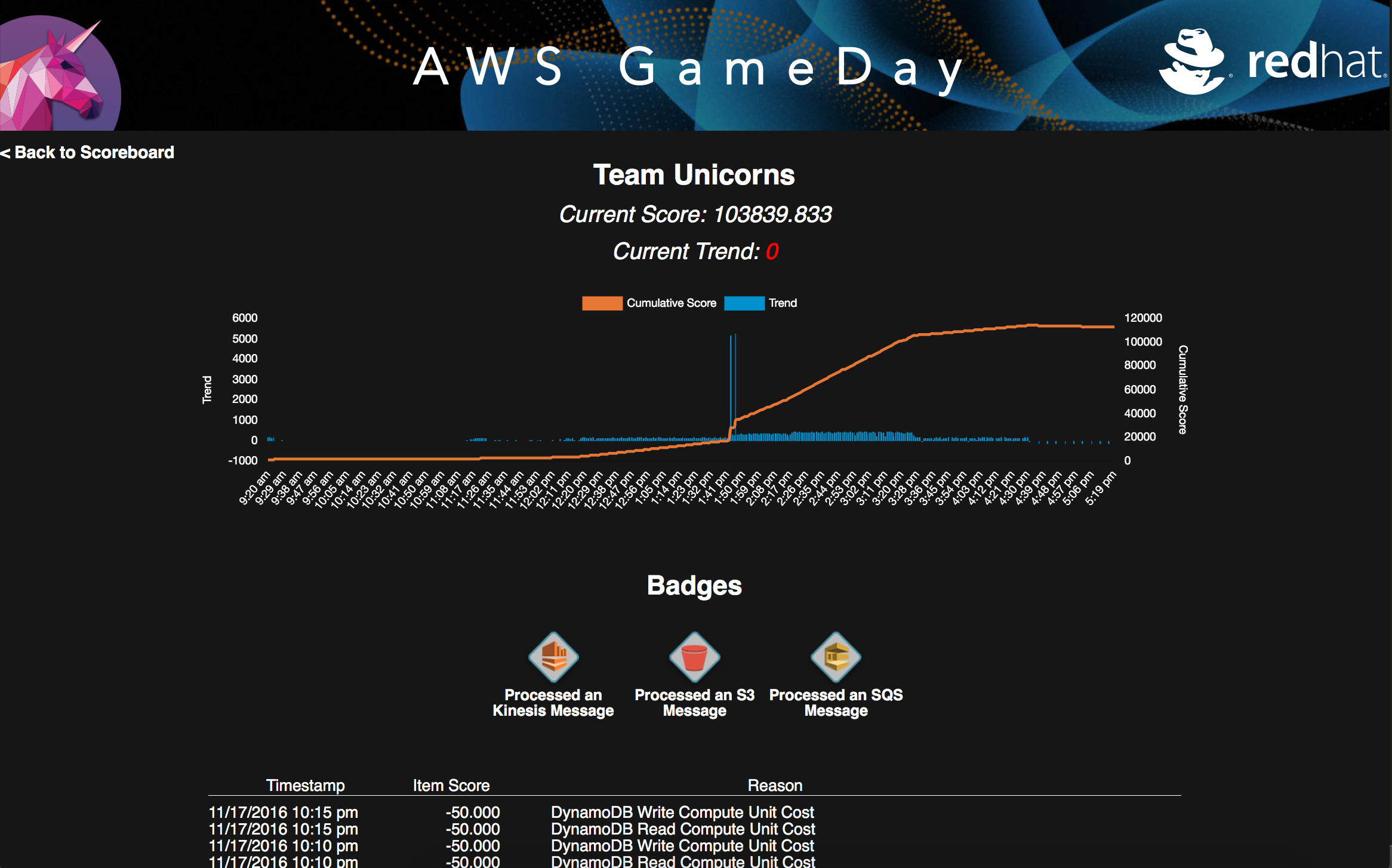
1. This button allows you to reach out for help to one of the Unicorn Rentals staff. When you click this button, someone will come to your table shortly to answer any questions you may have. If you need help immediately, feel free to stop by one of the helpdesks—both AWS and Ansible will be there.
2. This button will provide you with various links that will be helpful throughout the day. There are links to the main scoreboard, the player runbook, code snippets, side quests, etc.
3. This is your API token which you will need when you run your application. This ties your messages back to your team to earn you points. This will be one of the arguments for running the server application.
4. In order to access your account and interact with the AWS APIs you will need credentials. Clicking on this button will give you temporary credentials that you can use in your terminal window. Keep in mind these credentials expire after one hour. To get new credentials, just click the request credentials button again, and a fresh set of credentials will appear. Below that is a link to log into the AWS console.
5. The section below is where you can enter in the endpoint for your application. Here you should put the IP address of an EC2 instance or a DNS name. Keep in mind that you will need to add the protocol “http/https” or any ports outside of 80/443. Also, remove any trailing slashes.
6. The bottom section is used to enter codes that your team finds throughout the day. These can be redeemed for points! You’ll find these codes upon completing side quests, and other random places throughout the game. Be on the lookout!

# Scoreboard

If you click on “Useful Links” on the Player Dashboard, it will give you a link to the Scoreboard. The Scoreboard is where you will be able to track your progress as a team against all the players in the room.

The column on the far right is the Score Trend. This is a summation of all the score events for your team over the past minute. So for example, if you processed 10 messages and got 9 points each, but also got billed for one EC2 instance your trend would be 60. (10 x 9) – (1 x 30) = 60.



To get a deeper view on the performance of your individual team, click your team name. You will be brought to your individual team’s scoreboard page:

This page has four sections to pay attention to:

1. The top lists your Current Score Trend. This is again a summation of your total points over the past minute. For example, in the past 60 seconds, this team earned a total of 323 points.
2. Below that there is a histogram showing your team’s progress throughout the day at one minute intervals. The blue bars represent your score trend, and the yellow line indicates your cumulative score. This chart can highlight areas where you may be losing points or your progress has slowed down. For example, in this chart there are some areas where the blue bars are at 0 or below 0, which indicate there may be a problem with the infrastructure.
3. Underneath the histogram is the section that displays the badges your team has earned. You can earn badges for things you do throughout the day. These badges can be for good or bad things that you have done. Any badges you earn will add or remove points from your score.
4. Lastly, at the bottom you can see the individual score entries that are adding/subtracting from your score. You will see things like infrastructure costs, successfully processed messages, duplicate messages or incorrect messages processed, etc. Pay close attention here as it can highlight problems or help you see why you have a negative score trend.

# The Unicorn Rentals Messaging System

Unicorn Rentals includes a messaging system to connect the company’s real components (people, servers, ball pits, chowder taps, etc.) to mythical and imaginary assets (unicorns, hippogriffs, fairy dust, elves, gnomes, hope, etc.). This system is what you will be taking care of here as part of your job at Unicorn Rentals. The way this works is that parts of messages are sent to you in JSON format over a transport (right now we are only supporting HTTP(S), but that could change). The parts include a data payload and are identified with an ID, a part number, and the total number of parts in the message. The software you are maintaining combines the data parts of messages with the same ID in order and submits it back to an API in Unicorn Rentals. You need to identify yourself with an API key to be able to get credit for assembling the messages. The software already works and is totally fine, but here are some more details.

Each message you receive will be JSON formatted. This is the format with some types and comments:

{

“Id”: *string*,

“TotalParts”: *integer, // Nearly always 2*,

“PartNumber”: *integer, // The first part is 0*,

“Data”: *string*

}

You may receive messages out of order, so be ready for that. Typically, messages with the same ID are delivered within a minute or two of each other. Here is what the message processing program does:

1. Stores the message in a database. *TODO – this is in memory now, so we can’t run more than one copy of the server at once. Should probably use an external database.*
2. Checks to see if it has two parts with the same ID.
3. If it does, concatenate the data portions of the parts together in order (Part 0 + Part 1).
4. Post that concatenated string back to the API with your API key. If you don’t use your API key then we don’t know who you are so no one gets any points. What a bummer.
5. *TODO – record that this message ID was already processed so we don’t send it again*.

When running your client application, you will want to use the endpoint below to send your processed messages to and receive points:

https://dashboard.cash4code.net/score

If you don’t like the way the program works or the way it was written, this is fine. Feel free to rewrite it in whatever way makes the most sense for you.

Assembling parts of the messages and sending them back to us is the best way to earn a lot of points—I mean make a lot of money—for Unicorn Rentals. You earn the most points for a successful response that is returned to us quickly. The longer you take to respond, the less points the response is worth, but it is still worth something so if you find a bunch of messages hanging around you should send them in. If you send a message back to us and it is incorrectly assembled, you will lose points. If you send us a response to the same message more than once, you will also lose points.

Lastly, and maybe most importantly, the system will be sending you a constant stream of message parts. If you stop responding to messages because something broke, you will be missing out on points. Make it a priority to respond to every message and store the message if you can’t respond right away. Good luck!

# Ansible

We here at Unicorn Rentals strive to be on the cutting edge of technology so we went to DevOps everything. Our friend over at Cash4Code was talking about this thing Ansible so we thought we would take a shot at it. She said it provides a good option for configuration management due to its simplicity, agentless architecture, and ability to interact easily with your ever-changing, scaling, and dynamic AWS architecture.

Ansible can be installed on your Mac OSX or Linux laptop or server with pip. Running Ansible on Windows is currently not recommended for this event. To install Ansible do the following:

pip install ansible

If you get an error that you don’t have sufficient privileges, you may have to prepend that command with sudo.

Ansible has these playbooks - in YAML format - that can be created to have Ansible run tasks. In the dashboard, in Useful Links, we stored a file called facts.yml. The playbook contains something like this:

- name: gather facts about current infrastructure

hosts: localhost

connection: local

gather\_facts: false

tasks:

- name: get more ec2 instance facts

ec2\_remote\_facts:

region: 'eu-central-1'

register: instance\_facts

- debug: var=instance\_facts

- debug: var=instances

- name: gather facts about vpc

ec2\_vpc\_net\_facts:

region: 'eu-central-1'

filters:

vpc-id: "{{instance\_facts['instances'][0].vpc\_id}}"

register: vpc\_facts

- debug: var=vpc\_facts

- name: gather facts about vpc subnets

ec2\_vpc\_subnet\_facts:

region: 'eu-central-1'

filters:

vpc-id: "{{instance\_facts['instances'][0].vpc\_id}}"

register: vpc\_subnet\_facts

- debug: var=vpc\_subnet\_facts

In the dashboard, click Request Credentials/ Console URL. Copy the export lines into the terminal or a command line. After that, run:

ansible-playbook facts.yml

This script gathers facts about your VPC and the EC2 instance running in the AWS account.

We have been working on a playbook called site.yml that can then run task playbooks that can create ec2 launch configurations and autoscaling groups so we don’t have to launch ec2 instances and configure them by hand anymore. Kicking around the idea of having a load balancer as well so we can have a consistent DNS name and not have to worry if an IP changes because and EC2 instance goes down. We’re also playing around with creating DynamoDB tables and a few other services.

For more information about Playbooks and how to use them, see the [Ansible documentation](http://docs.ansible.com/ansible/intro.html)

Let’s automate all the things!!