

https://s3.amazonaws.com/ee-assets-prod-us-east-1/modules/gd2015-loadgen/v0.1/docs/runbook.md

**Documentation** 

## respond well to simple horizontal scaling. As there is expected to be a constrained set of client keys, the system should respond well to caching techniques. That's it for hints. Scope

Here at Unicorn Rentals we know that it takes money to make money. You will be charged \$8

"Set Team Name" button on the dashboard and enter a team name. Keep in mind that this team

**TREND** 

633.67 103.67 1

Modules

Mini Games

Game

Scoreboard

Dashboard Guide

**AWS Console** 

**RANK** 

Chat Invite

Score: 1,027 / Trend: 169

Logout

1. The top bar of the dashboard has a series of buttons that allow you to: 1) Access your score events, 2) Access the scoreboard, 3) Set your team name (can only be done once), 4) Request invite to Slack channel for team

1. Each row lists every score event that your team has generated. The "Source" column tells you where the point awards or deductions came from. The "Points" column will tell you how many points you got or lost.

2. The "Reason" column will tell you the reason you got the points or lost the points or lost the points. Pay very close attention to this column when you are losing points in order to understsand what is going on and how to fix the problems.

name cannot be changed once set and will represent your team on the scoreboard. The

Score Events

Current Value: http://52.72.237.43

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

If you click on the "Scoreboard" link on the Player Dashboard, it will open a window 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.

**SCORE** 

Game ID: 41745e799a124ad9b6215d6c7c539c00 Team ID: c53d911fb7af465e997906ab5dfcfd6e

scoreboard team at Unicorn.Rentals did not have the motivation to make the scoreboard

for every 15 minutes of EC2 usage. For example, if you have one EC2 instance running for

3 minutes and terminate it, you will be charged \$8. If you have 2 instances running for

### This runbook describes the operational theory and practice for the production system powering the Unicorn Rentals website. The primary audience is the DevOps team running the site. The DevOps team is responsible for deploying code, scaling the site in response to load, maintaining our published SLA's (including response time and uptime), disaster

# recovery, troubleshooting activities and any monitoring and alerting activities required to meet these objectives.

**Infrastructure Cost** 

13 minutes, you will be charged \$16. So make sure you only provision what you need! Player Dashboard The dashboard can be accessed by going to https://dashboard.eventengine.run. It will prompt you to enter your Team Hash. This team hash can be found on a piece of paper handed to you earlier today. Once you login to the dashboard, it's time to set your team name. Please click on the

## private. It is a public site and team names should be created with that in mind. Any inappropriate team names will be renamed by the Unicorn. Rentals board of directors. Dashboard

Game: gd2015-doc-test

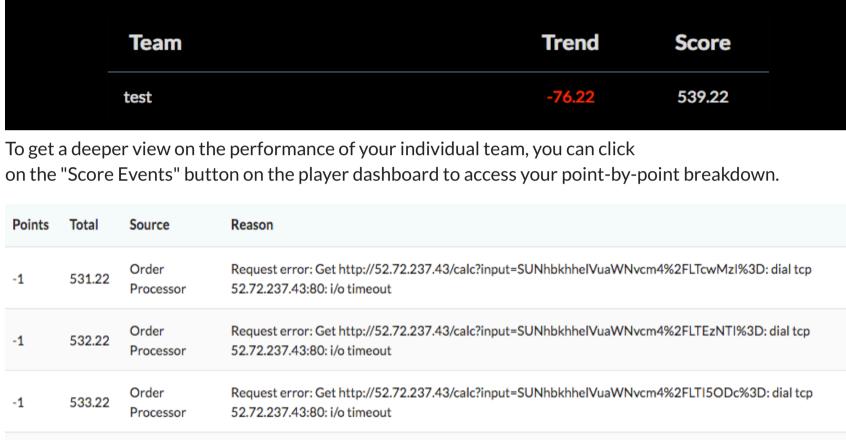
# **Order Processor** Outputs: No outputs defined Inputs: Server Address

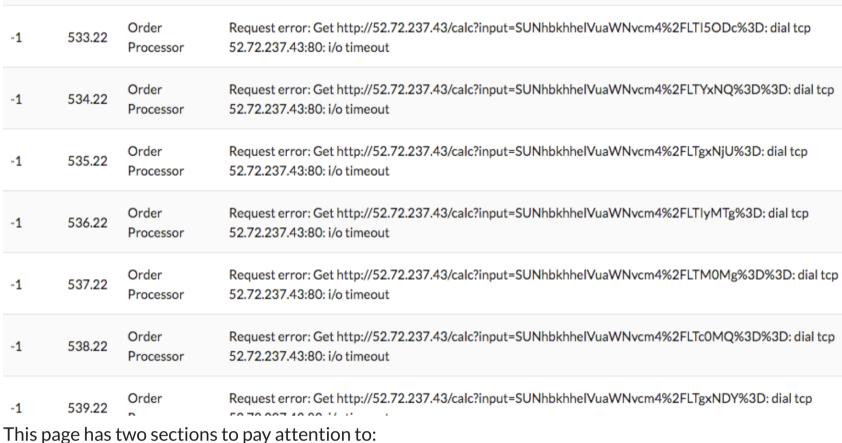
Team Name: test

Update Can Kyle Sandiego order unicorns by text from anywhere in the world? todo Instructions Answer

## collaboration and chatting, and 5) Access your team's AWS account (<-- this is important!) 2. Below this bar you will see a list of modules enabled for this game. Each module can define outputs and inputs. In the example above, the module has no outputs, but does have one input. For this module, you will provide a URL to your order processor so we can send you order (and you can get points). NOTE: Once you set your team name, and your team name is approved by the UnicornRentals collective, additional options will appear on the dashboard. These options include your team score, rank, and trend, as well as access to the scoreboard via the "Score Events" button. Score Events and Scoreboard

AWS GameDay





## 1. The server application is deployed, as a 'go' binary compiled from source rumored to be stored in a github repository. However, the name of of this repository is unknown to the current operations and development staff. 2. The server application can handle about 5 connections before starting to get really slow. Be careful about overloading, and watch for 503s when the queue fills. You can try restarting the app or server if it starts to get backed up. 3. The server application is an x86 statically linked, unstripped ELF executable found here: https://s3.amazonaws.com/ee-assets-prod-us-east-1/modules/gd2015-loadgen/v0.1/server 4. The base OS we have chosen is Amazon Linux. This distribution was selected for it's broad industry support, stability, availability of support and excellent integration with AWS. This distributions was selected by SecOps based on their requirements for platform hardening. 5. Architecture was moved to AWS as part of go-to-market plan. Operating the AWS CLI (http://docs.aws.amazon.com/cli/latest/userguide/installing.html) might be helpful. endpoint\_url

Elastic Load Balancer

Amazon EC2

http://endpoint\_url/

Return Hash

• Our Server code is downloaded on first-boot via a script provided by AWS as 'User Data'

• \$ aws autoscaling describe-launch-configurations --query LaunchConfigurations[0].UserData | base64 --decode

• The User Data is stored in the Auto Scaling Group 'Launch Configuration':

**GET Object** 

Server

'server' process

1. Make a new Launch Configuration based on the old one (http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/WorkingWithLaunchConfig.html)

References

AMI Unicorn\_High\_Level\_AWS 1. When working with AWS, only the following roles are allowed by SecOps... and finance:

> o ec2 o s3

ebs ecs eks cfn

elasticache

## cloudwatch o sns systems manager cloudtrail config o vpc cloudfront lambda 2. Getting bored of writing this silly thing. Who needs it? **Application System Architecture** Client <-> Server. Pretty sure anyway. What else is needed here? Client

# Unicorn\_Data\_Flow **On-line Operations Update Application**

## wget \*binary location (reference 3)\* chmod +x server ./server # Reboot if the server crashes shutdown -h now Here is an overview of the AutoScaling Group Launch Configuration:

Auto Scaling

**Auto Scaling** Launch Configuration

#!/bin/bash

# ш

3. Yes, this means you must relaunch the instances. **Update Auto Scaling Policy** Never tried this! Check out http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/scaling\_typesof.html **Troubleshooting Procedures Networking walkthrough** The AWS VPC / ELB environment must be healthy for the application to work. All production traffic flows through the ELB on both ingress and egress:

2. Update the location of the server code downloaded from S3 in 'User Data' section.

5. Make sure the instance can see the S3 bucket on launch (default route via IGW in subnet).

1. Create an ssh-keypair (http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html)

3. Associate this new launch configuration with the Auto Scaling group.

2. Update Launch Configuration (Same procedure as 'Update Application')

4. Scale-up new group / scale-in old group

Elastic Load

Add ssh-key to instance

## **Auto Scaling Group** Security Group A Subnet A Subnet B Region us-west-2 Unicorn\_Detail\_Network 1. Check the Security Group settings for your instances 1. Make sure all required ports are Allowed 2. Check the Routing tables on your subnets 1. Make sure the routing tables are applied to each subnet 2. 'Default' table applies to all subnets without an explicit definition 3. Make sure the routing table has the appropriate rules 3. Things to check in the the VPC that Ops has broken in the last six - 12 months. 1. Instances up? Try "http://<<Your-IP-or-DNS>>/healthcheck" to for more detail 2. Instance 'up' in the Auto Scaling group?

3. Subnets? 1. Subnet 'cider'??? What did that guy say it was called? Is it big enough??? 2. Are the subnets added to the Elastic Load Balancer? 3. Are the subnets added to the Auto Scaling Group? 4. Routes correct / intact? See diagram. 5. ACLs: set on subet. Too restrictive/permissive? 6. Security groups? 7. IGW? Do we have routes to flow traffic through the IGW? Required to grab the server code from S3. 8. Route53 records? Are the records pointing to the correct resources? • New application test utility! https://ee-assets-prod-us-east-1.s3.amazonaws.com/modules/gd2015-loadgen/v0.1/server-bang.linux https://ee-assets-prod-us-east-1.s3.amazonaws.com/modules/gd2015-loadgen/v0.1/server-bang.osx https://ee-assets-prod-us-east-1.s3.amazonaws.com/modules/gd2015-loadgen/v0.1/server-bang.windows

./server-bang.\$arch <server URL> • You can try ssh'ing to the node and checking out the application -it runs on port 80: thats about all I know. You must install an ssh key first (see 'Add ssh-key to instance', above) • As mentioned, the server process can get slow if it is handling too many connections. Try restarting. **Change Management** HAH! **System Monitoring** 

http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/policy\_creating.html

• Explore elasticache, cloudfront: Didn't Jayme say this might speed things up?

http://docs.aws.amazon.com/ElasticLoadBalancing/latest/DeveloperGuide/elb-cloudwatch-metrics.html

• Application monitoring: https://scoreboard.eventengine.run

• Refactor infrastructure using container architecture

How to check ELB metrics?

• How to test HA? Automate DR.

To Do