

Test Project Main document

Cloud Computing

Submitted by: Taze Miller US

Contents

This Test Project consists of the following documentation/files:

- WS2019_TP53_Main_document
- WSC2019_TP53_DayOne_actual
- **WSC2019_TP53_DayTwo_actual**
- WSC2019_TP53_DayThree_actual
- WSC2019_TP53_DayFour_actual

Introduction

In recent years, Cloud Computing has become a necessity for businesses across all sectors and verticals. To keep a competitive edge, businesses leverage the cloud to develop solutions that can handle the demands of their customers and give them a positive customer experience given any load or fault scenario. There are key aspects to successfully building a cloud-based solution. These include system design, deployment, network design, high availability, scalability, automation, security, cost, and monitoring. This test project will assess Competitors based on their ability to effectively and securely deploy, maintain, and scale a web application.

These aspects are explained in more detail in the “Scaling A Web Application Break Down” section below.

Scope

This document describes the operational theory and practice for the production system powering the Unicorn Rentals website. The primary audience is the Unicorn Rentals DevOps team running the site. This team is responsible for deploying code, scaling the site in response to load, maintaining published SLAs (including response time and uptime), disaster recovery, troubleshooting activities, and any monitoring and alerting activities.

Tasks

1. Log into Gameday with your assigned hash (Provided on the day)
2. Set your team/competitor name on the Dashboard – (Format: NAME SURNAME)
3. Read the documentation thoroughly (Outlined below)
4. Log into the AWS console (link provided from the Dashboard)
5. Examine existing configurations in EC2 (Elastic Cloud Computer server)
6. Examine existing configurations in VPC (Virtual Private Cloud, Network Segment)
7. Configure application to auto scale to handle increasing load (Auto Scaling Groups, with Launch Configuration)
 - (a) Create/update the user data configuration correctly, including download locations of server binary and server configuration file
 - (b) Update the user data configuration to include any required local dependencies
8. Configure any server dependencies as outlined in the technical details
9. Configure necessary application monitoring, metrics and alarms in CloudWatch
10. Monitor performance of the application servers in the “Score Events and Scoreboard” and through the AWS Console with CloudWatch

11. Serve client requests to gain points, reference the “Score Events and Scoreboard” to ensure you are scoring positively by serving the requests.
12. Monitor costs and do not scale up the infrastructure excessively to minimize penalties
13. Process exceptions when they are received, reference the “Request Exception Handling” below.

Initial state Days 1-2

At the start of the day, the minimal infrastructure needed to provide the public client API will be operational within the account. This infrastructure is neither scalable nor highly available. It does however provide a reference point for a functional deployment.

Please reference the TeamRole your account IAM console for any permissions based questions.

Summary

[Delete role](#)

| | |
|---|--|
| Role ARN | arn:aws:iam:: [REDACTED] :role/TeamRole Copy |
| Role description | Edit |
| Instance Profile ARNs | arn:aws:iam:: [REDACTED] :instance-profile/TeamRoleInstanceProfile Copy |
| Path | / |
| Creation time | 2019-08-21 03:17 CDT |
| Maximum CLI/API session duration | 12 hours Edit |
| Give this link to users who can switch roles in the console | https://signin.aws.amazon.com/switchrole?roleName=TeamRole&account-id=[REDACTED] Copy |

Permissions
Trust relationships
Tags
Access Advisor
Revoke sessions

▼ Permissions policies (3 policies applied)

[Attach policies](#)
[+ Add inline policy](#)

| Policy name ▼ | Policy type ▼ | |
|--|----------------|---|
| ▶ restrict-policy | Managed policy | ✕ |
| ▶ ws-loadgen-aws-day2-policy | Managed policy | ✕ |
| Show 1 more | | |
| ▶ Permissions boundary (not set) | | |

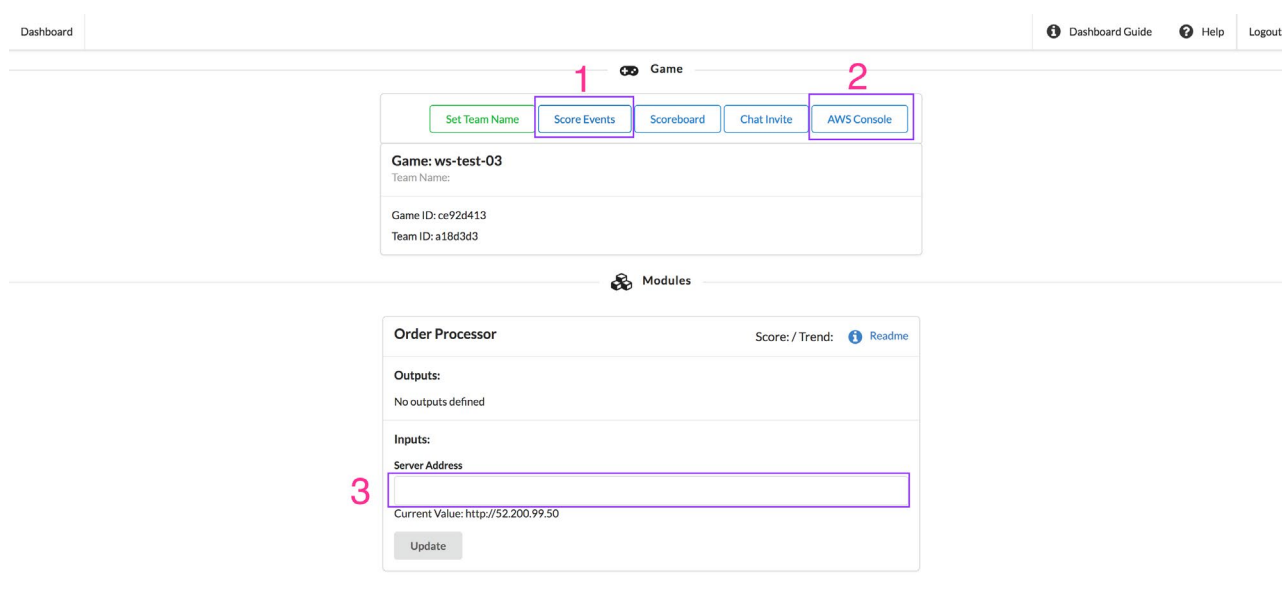
Infrastructure Cost

Scaling the infrastructure horizontally brings an increase in the number of Amazon EC2 instances used. Each instance is associated with a cost. If there are more instances deployed than necessary to meet the demand of the load, Competitors will be penalized in points. Make sure to deploy the necessary number of instances to meet the demand (which changes throughout the day).

Personal Event Dashboard

The dashboard can be accessed by going to <http://dashboard.eventengine.run/>. It will prompt you to enter your team/competitor hash. This hash can be found on a piece of paper handed to you earlier today.

The personal event dashboard and scoreboard is provided to give Competitors some visibility into how their application is performing based on web traffic served. This dashboard, however, *does not* include the Marks that are given based on Systems Design and Deployment, Systems Design and Deployment, Network Design and Deployment, Infrastructure Automation, Infrastructure Security, Infrastructure Active and Passive Monitoring. Each Criteria will provide marks that will be added up to meet the total amount. The sole purpose of the Personal event dashboard and Scoreboard for Competitors to have visibility into viability and how they are serving traffic and is not how the Competitors are performing in relation to other based on all the Criteria's where Marks can be accorded.



The screenshot shows the 'Personal Event Dashboard' interface. At the top, there is a navigation bar with 'Dashboard', 'Game', 'Dashboard Guide', 'Help', and 'Logout'. Below this, a 'Game' section contains several buttons: 'Set Team Name', 'Score Events' (annotated with a red '1'), 'Scoreboard', 'Chat Invite', and 'AWS Console' (annotated with a red '2'). Below the buttons, the game details are displayed: 'Game: ws-test-03', 'Team Name:', 'Game ID: ce92d413', and 'Team ID: a18d3d3'. Below the game details, there is a 'Modules' section. The first module is 'Order Processor', which has a 'Score: / Trend: Readme' link. It shows 'Outputs: No outputs defined' and 'Inputs: Server Address'. A text input field for 'Server Address' is highlighted with a red box and a red '3'. Below the input field, the 'Current Value: http://52.200.99.50' is displayed, and there is an 'Update' button.

The dashboard has a few key components that you will interact with throughout the competition. The top bar of the dashboard has a series of buttons that allow you to:

1. Access your score events. These are individual entries of activity helpful in determining the availability of your application.
2. Access your AWS account. Click on this button in order to get access to your AWS account. You are provided with an AWS account to use for this competition. On completion of each day, the account will be closed and unable to be accessed again.
3. Input for your infrastructure address. Your web infrastructure will require a public hostname to access. Just as every website requires a name in a web browser, users of this web application you are deploying require an address to use in order to access the site. You can use an IP address for this address but will get more points for a hostname address.

Score Events and Scoreboard

To get a deeper view your performance, you can click on the "Score Events" button on the player dashboard to access your point-by-point breakdown.

| Points | Total | Source | Reason |
|--------|--------|-----------------|---|
| -1 | 531.22 | Order Processor | Request error: Get http://52.72.237.43/calc?input=SUNhbkhhelVuaWNvcm4%2FLTcwMzI%3D: dial tcp 52.72.237.43:80: i/o timeout |
| -1 | 532.22 | Order Processor | Request error: Get http://52.72.237.43/calc?input=SUNhbkhhelVuaWNvcm4%2FLTEzNTI%3D: dial tcp 52.72.237.43:80: i/o timeout |
| -1 | 533.22 | Order Processor | Request error: Get http://52.72.237.43/calc?input=SUNhbkhhelVuaWNvcm4%2FLTISODc%3D: dial tcp 52.72.237.43:80: i/o timeout |
| -1 | 534.22 | Order Processor | Request error: Get http://52.72.237.43/calc?input=SUNhbkhhelVuaWNvcm4%2FLTYxNQ%3D: dial tcp 52.72.237.43:80: i/o timeout |
| -1 | 535.22 | Order Processor | Request error: Get http://52.72.237.43/calc?input=SUNhbkhhelVuaWNvcm4%2FLTgxNjU%3D: dial tcp 52.72.237.43:80: i/o timeout |
| -1 | 536.22 | Order Processor | Request error: Get http://52.72.237.43/calc?input=SUNhbkhhelVuaWNvcm4%2FLTyMTg%3D: dial tcp 52.72.237.43:80: i/o timeout |
| -1 | 537.22 | Order Processor | Request error: Get http://52.72.237.43/calc?input=SUNhbkhhelVuaWNvcm4%2FLTM0Mg%3D: dial tcp 52.72.237.43:80: i/o timeout |
| -1 | 538.22 | Order Processor | Request error: Get http://52.72.237.43/calc?input=SUNhbkhhelVuaWNvcm4%2FLTc0MQ%3D: dial tcp 52.72.237.43:80: i/o timeout |
| -1 | 539.22 | Order Processor | Request error: Get http://52.72.237.43/calc?input=SUNhbkhhelVuaWNvcm4%2FLTgxNDY%3D: dial tcp 52.72.237.43:80: i/o timeout |

This page has two sections to note:

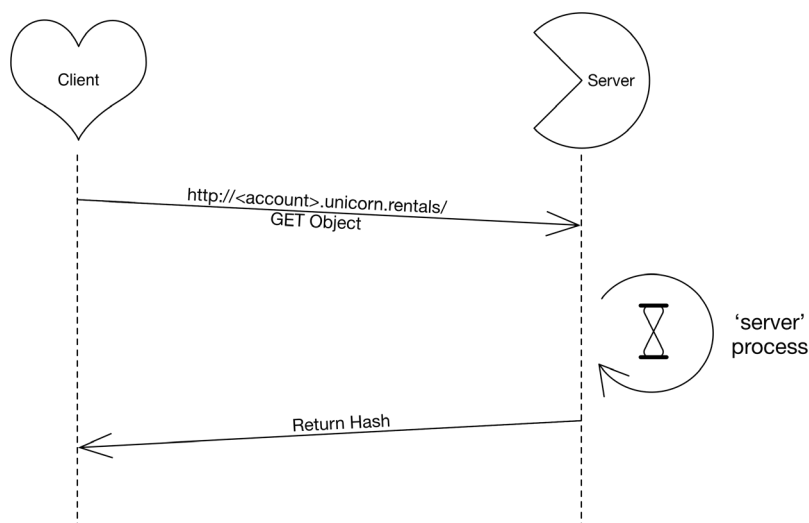
1. Each row lists every score event that you have generated. The "Source" column tells you where the point awards or deductions came from. The "Points" column will tell you how many points you have received or lost.
2. The "Reason" column will tell you the reason you received the points or lost the points. Pay very close attention to this column when you are losing points in order to understand what is going on and how to fix the problems.

AWS Services

When working with AWS, you have access to most services. If you get an error such as "Permission Denied", check to make sure that you are operating in the correct AWS Region and using appropriate resources sizes (e.g. "t2" instance sizes).

Application Architecture

The client below is making requests to the server that you are running. The address the client tries to connect on is defined by the address that you set in the dashboard.



On-line Operations

You can find additional application information on options by using the "-h" flag when running the server binary:

```
./server -h
```

1. Create a new Launch Configuration based on the existing Launch Configuration that can be found in the account provided:
2. (<http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/WorkingWithLaunchConfig.html>)
3. Utilize a 'User Data' script
4. Associate this new launch configuration with the Auto Scaling group.
5. Scale-up new group / scale-in old group.
6. Make sure instances are associated with the proper network resources.

Add an ssh-key to the instance

1. Create an ssh-keypair (<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html>)
2. Update the Launch Configuration (Same procedure as 'Update Application').
3. Yes, this means you must relaunch the instances.

Connect from Windows: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>

Configure an Auto Scaling Policy:

http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/scaling_typesof.html

Request Exception Handling

Throughout the time your web applications are running, you will occasionally receive a request in the application log file and console of the running application indicating that an exception has occurred. An example of this is shown below:

04-22-2018:12:12:12 Warning: Unicorn Refund Request: c4a5010e-6734-41e8-974b-59af1bd55ca0

Throughout the day at random intervals you will receive a request similar to the one identified above. Each request will have a string associated in a UUID format. Each of these requests identifies a customer request for a refund on a Unicorn Rental purchase. Each of these requests must be sent to the endpoint provided. Every time one of these requests is received, you should send it to the audit system using an HTTP POST call and the tool of your choice. The example below demonstrates using the utility "curl" on the Linux command line:

```
curl -i -H "Accept: application/json" -X POST -d '{"game": "GAME_ID", "team": "TEAM_ID",  
"order": "REQUEST_UUID"}' https://stats.aws.dev-null.link/proc/refund
```

There is also a script that can be used. This script is downloadable at:

<https://s3-us-west-2.amazonaws.com/ws-bucket-data/assets/uc-refund.sh>

To use this script, you would just perform the following action:

```
user@host > chmod 755 uc-refund.sh  
user@host > ./uc-refund.sh GAME_ID TEAM_ID REQUEST_UUID  
Request for refund in review!  
user@host >
```

The three needed items are:

GAME_ID - Found at the top of your player dashboard

TEAM_ID - Found in the middle of your player dashboard

REQUEST_UUID - This is the long string of characters in the message. In the example above, this is, "c4a5010e-6734-41e8-974b-59af1bd55ca0"

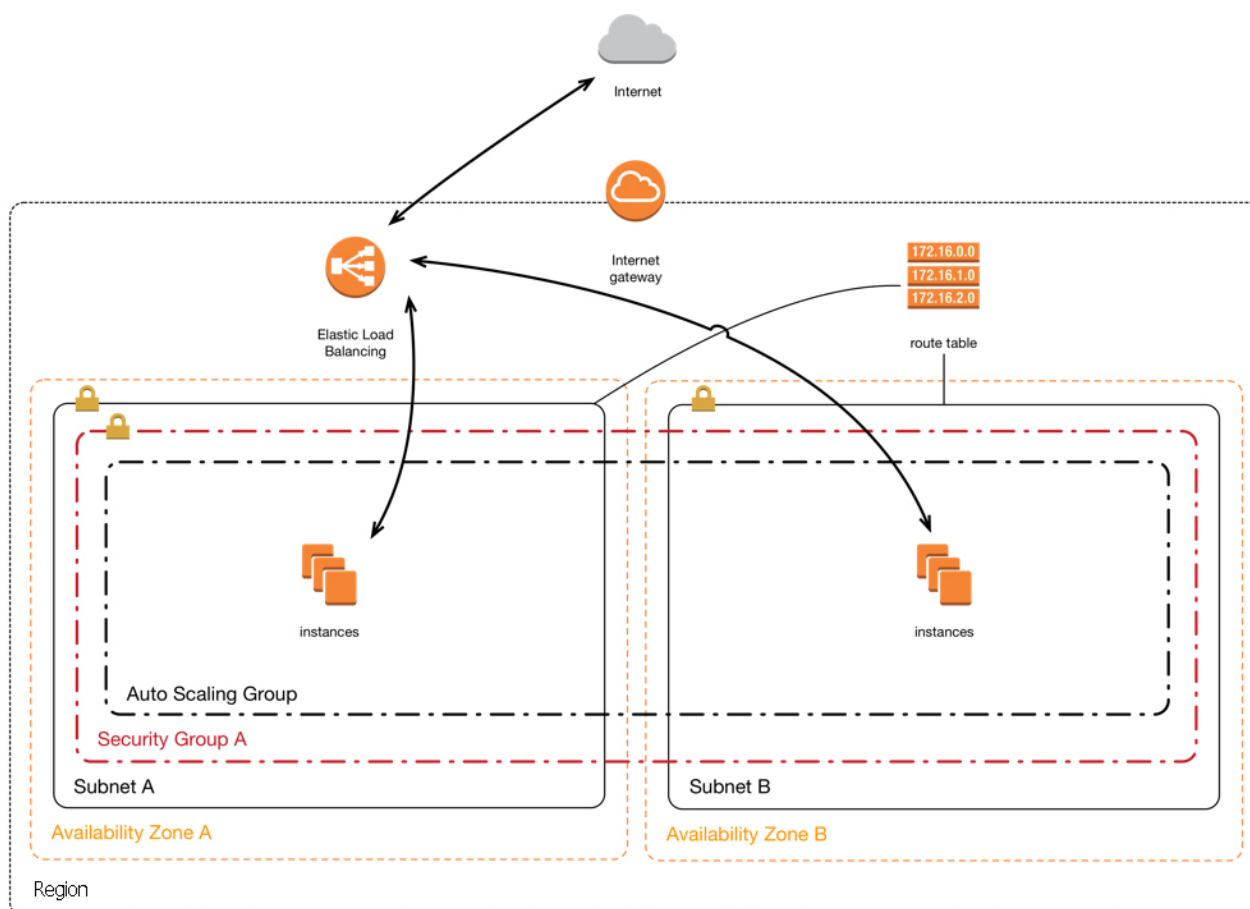
NOTE: You may submit the same REQUEST_UUID multiple times without penalty. You will however only receive credit on the first submission of each unique value. Duplicate submissions will be ignored.

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 as shown in this diagram.



1. Check the Security Group settings for your instances
 - (a) Make sure all required ports are allowed
2. Check the Routing tables on your subnets
 - (a) Make sure the routing tables are applied to each subnet
 - (b) 'Default' table applies to all subnets without an explicit definition
 - (c) Make sure the routing table has the appropriate rules
3. Things to check in your VPC.
 - (a) Are the Instances up?
 - (b) Is the Instance 'up' in the Auto Scaling group?
 - (c) Are your subnets configured properly?
 - (i) Subnet details and size are an important component
 - (ii) Are the subnets added to the Elastic Load Balancer?
 - (iii) Are the subnets added to the Auto Scaling Group?
4. Are Routes correct / intact? See the above diagram.
5. Are ACL set on subnet? Are they too restrictive/permissive?
6. Are you using the correct Security groups?

7. Internet Gateway (IGW) Do you have routes to flow traffic through the IGW? Required to grab the server code from S3.
8. DNS settings: Are the records pointing to the correct resources?
9. You can try connecting to the instance using SSH to verify the server application is working correctly and to access the application logs. You must install a ssh key first (see 'Add ssh-key to instance', above)
10. Performance: The server process can get slow if it is handling too many connections. Try restarting the server if it becomes overloaded.
11. Security consideration: you will have created a configuration file containing database credentials and other sensitive data. Is this something that you want available for public download?

Application testing

Accessing the healthcheck endpoint at `http[s]://[my-endpoint]/healthcheck` will help determine if the server is functional and display the current load as follows:

Current outstanding tasks: #

You can also find a test utility below available for the architecture of different clients. Please note that this is a linux binary and the filename may differ from the example below:

<URL TO UNICORN TESTER APP UTILITY WILL BE PROVIDED ON THE DAY>

Running the test utility

```
./server-bang.$arch <server URL>
```

Application Versions

During the course of the test, you will be required to "update" your application. A new version of the application will be made available to download. Once it is available, you will want to start rolling out that new version. At the defined time, the old version of the application will no longer function.

System Monitoring

How to check ELB metrics?

http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/policy_creating.html

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

Scaling A Web Application Break Down

Systems Design/Deployment – When designing and deploying a web application, the fundamental building blocks of being able to scale is understanding how to implement an architecture that can scale. Competitors will need to showcase their understanding in decoupling the database from the application, utilizing additional options and effective implementation of integration.

Network Design – When scaling a web application and breaking up the workload into different tiers and services, the network design must ensure that only servers and services that should be public remain public. To ensure network security, the application should communicate with services on private networks where possible.

High Availability – In today's web applications high availability is an essential aspect. Competitors will need to keep this in mind and implement ways to ensure the web application can deal with issues and still remain a running application.

Scalability – In order to keep costs low when there is low usage and scale to meet high traffic to provide a consistent user experience, the application must scale or the application must be scalable.. Scalability in every aspect of the web application allows the application to grow only where needed. Correctly implemented this goes hand in hand with monitoring and automation.

Automation– Automation is one of the fundamental building blocks of being able to scale a web application. Automation of application deployment process, infrastructure provisioning automation and self-configuration.

Security – When scaling a Web Application, security at every layer of the application is essential. Where network traffic is allowed to come from, who can access the servers, what permissions are applied to the servers and users, who has access to the databases and data. Security can be applied on every aspect of a Web application.

Monitoring – Monitoring has become the most important aspect of a web application. Being able to collect metrics and understand how the web application is behaving at all layers. Being able to use those metrics to scale up and down and use those metrics to make smart decisions and automation where possible

Links

<https://www.youtube.com/watch?v=vg5onp8TU6Q> - Scaling Up to Your First 10 Million Users

https://d1.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf

<https://aws.amazon.com/blogs/startups/scaling-on-aws-part-1-a-primer/>

<https://aws.amazon.com/blogs/startups/scaling-on-aws-part-2-10k-users/>

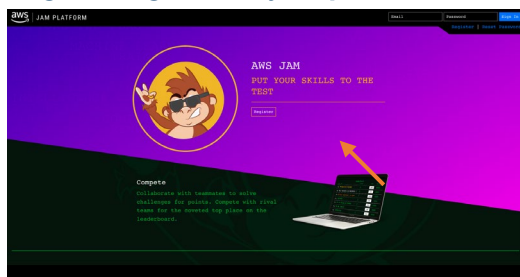
<https://aws.amazon.com/blogs/startups/scaling-on-aws-part-3-500k-users/>

<https://aws.amazon.com/blogs/startups/scaling-on-aws-part-4-one-million-users/>

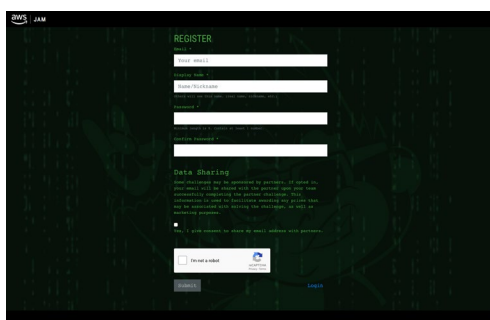
Additional Test Modules Through the Jam Platform

The first two days of competition will be mainly focused on your ability to create new scalable architectures. In addition to testing your creativity and technical ability in creating architectures, you will also be tested on specific skills that are necessary as a cloud computing expert. For this part of the competition we will be using a module-based platform for testing specific skills.

Registering on the jam platform



If you haven't already just click the register button.



Fields:

Email: Your Email

Display Name: [Name]-[Country]

Password: A password of your choosing. Never share this with another person

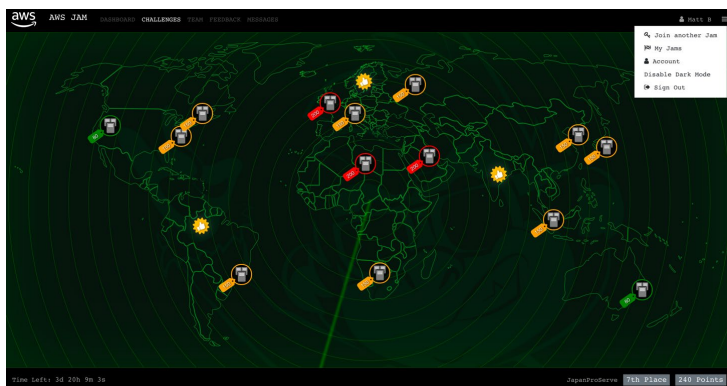
You can choose whether or not to give consent to distribute your email. For this event it will not matter.

Complete the Captcha

Using the Jam platform

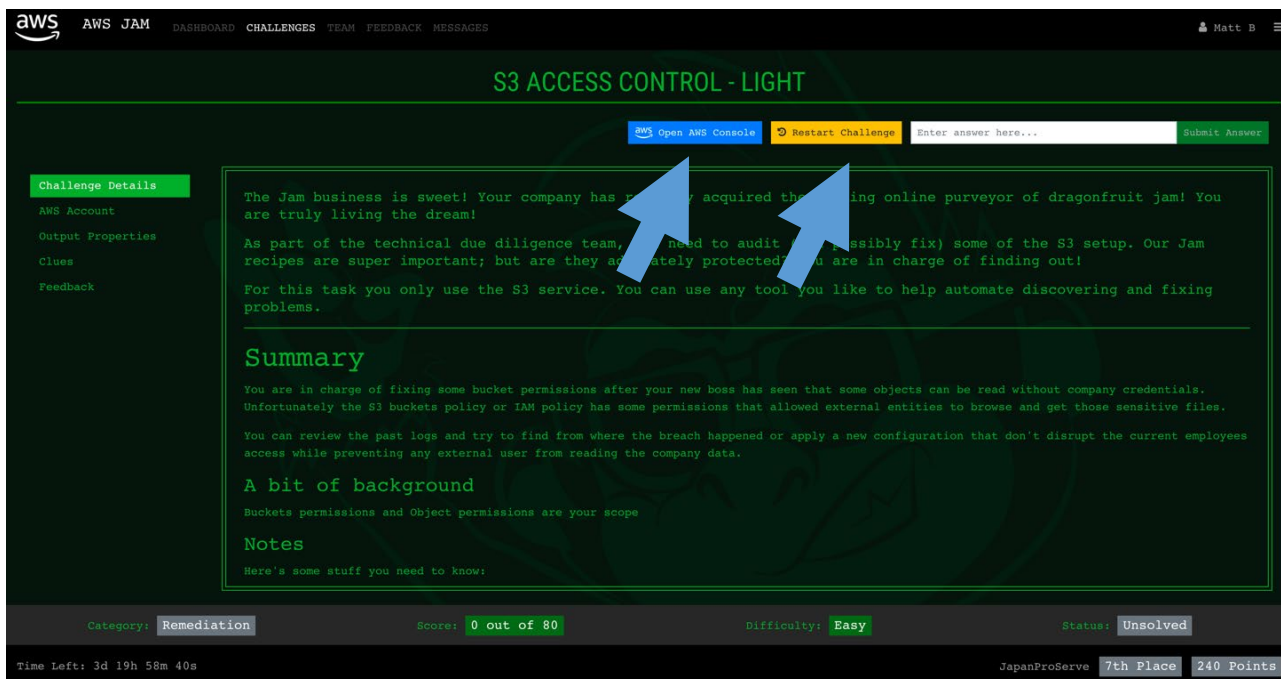
You will be using the jam platform every day for the competition. Days 1 and 2 will be small modules that will open towards the end of the day. Days 3 and 4 will be exclusively jam platform based. Once the jam event opens, you'll be able to complete the challenges in any order that you wish.

Once we are ready to start the jam event, you will all be given an event password. This will unlock your tasks for that day.



The default view of the jam platform is a map. This is just a fun layout, but is not important for this event. You can complete any challenge in any order.

Once you select a challenge you will see a screen like this:



S3 ACCESS CONTROL - LIGHT

[AWS Open AWS Console](#) [Restart Challenge](#) Enter answer here... [Submit Answer](#)

Challenge Details

- AWS Account
- Output Properties
- Clues
- Feedback

The Jam business is sweet! Your company has recently acquired the leading online purveyor of dragonfruit jam! You are truly living the dream!

As part of the technical due diligence team, you need to audit (and possibly fix) some of the S3 setup. Our Jam recipes are super important; but are they adequately protected? You are in charge of finding out!

For this task you only use the S3 service. You can use any tool you like to help automate discovering and fixing problems.

Summary

You are in charge of fixing some bucket permissions after your new boss has seen that some objects can be read without company credentials. Unfortunately the S3 buckets policy or IAM policy has some permissions that allowed external entities to browse and get those sensitive files.

You can review the past logs and try to find from where the breach happened or apply a new configuration that don't disrupt the current employees access while preventing any external user from reading the company data.

A bit of background

Buckets permissions and Object permissions are your scope

Notes

Here's some stuff you need to know:

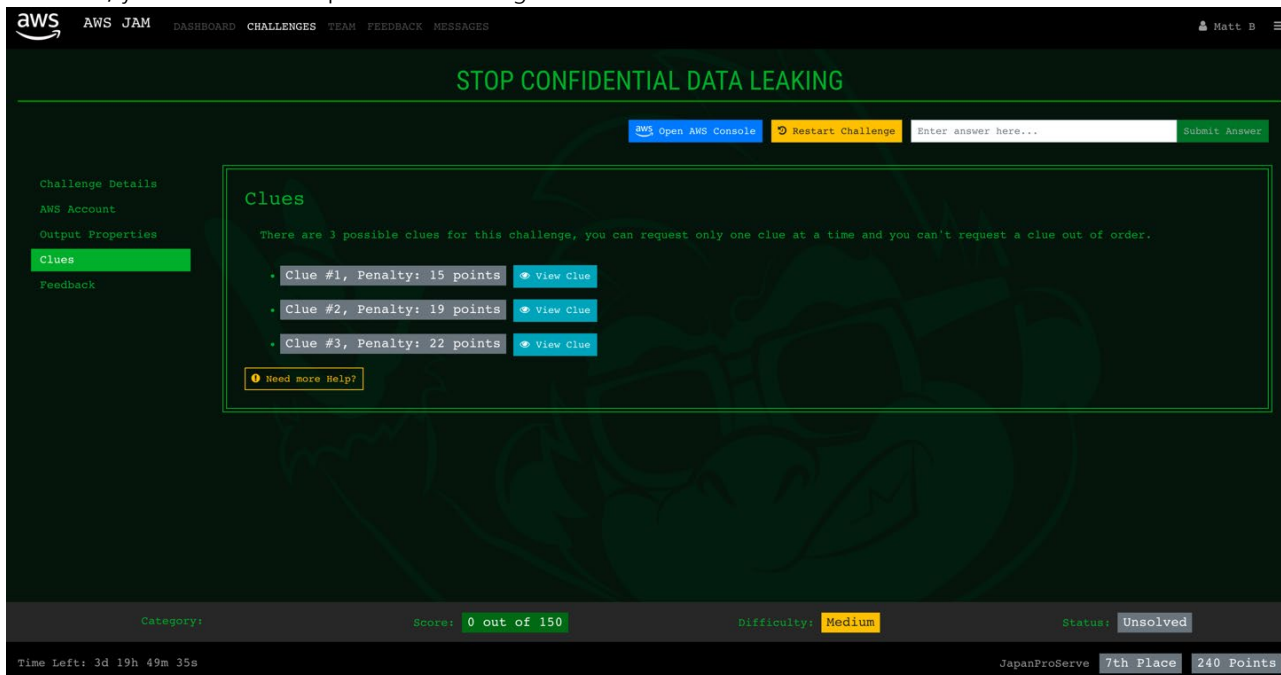
Category: Remediation Score: 0 out of 80 Difficulty: Easy Status: Unsolved

Time Left: 3d 19h 58m 40s JapanProServe 7th Place 240 Points

This will give you the instructions for the task, access to your account, and the ability to restart the challenge if you believe you've broken something and cannot repair it yourself.

clues

The jam modules can be very difficult, but all can and have been solved using only the instructions given to you. However, you do have the option of revealing clues.



STOP CONFIDENTIAL DATA LEAKING

[AWS Open AWS Console](#) [Restart Challenge](#) Enter answer here... [Submit Answer](#)

Challenge Details

- AWS Account
- Output Properties
- Clues
- Feedback

Clues

There are 3 possible clues for this challenge, you can request only one clue at a time and you can't request a clue out of order.

- Clue #1, Penalty: 15 points [View Clue](#)
- Clue #2, Penalty: 19 points [View Clue](#)
- Clue #3, Penalty: 22 points [View Clue](#)

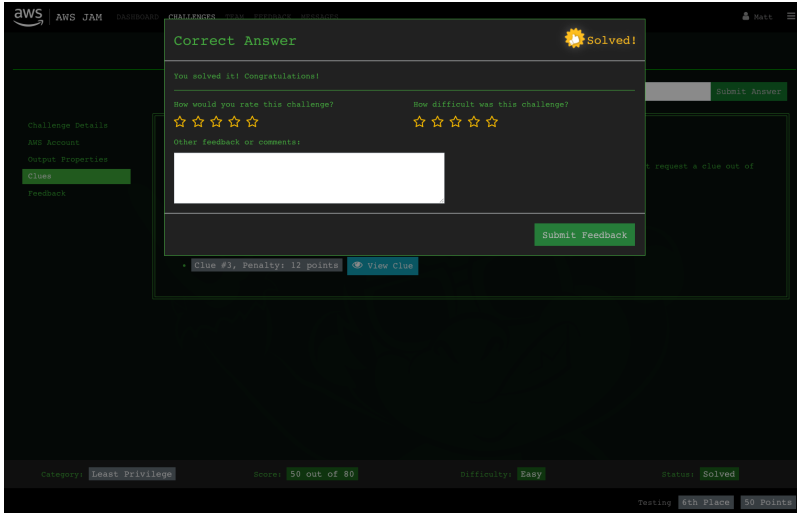
[Need more Help?](#)


Category: Difficulty: Medium Status: Unsolved

Time Left: 3d 19h 49m 35s JapanProServe 7th Place 240 Points

Clues are designed to help you make progress toward the final solution but it is important to remember that **you will not receive full points for the module if you use a clue**. Determine your own strategy, and be mindful of the consequences of using a clue.

After you complete each challenge a feedback page will be displayed:



Correct Answer 

You solved it! Congratulations!

How would you rate this challenge? ☆☆☆☆

How difficult was this challenge? ☆☆☆☆

Other feedback or comments:

Clue #3, Penalty: 12 points

Category: **Least Privilege** Score: **50 out of 80** Difficulty: **Easy** Status: **Solved**

Testing **6th Place** **50 Points**

Please answer honestly as time permits. We will use this feedback to influence future competitions.

As always **Good Luck!**