Game Kingdom
Final Report

Key Contributors: Joseph Drouillard, Anthony Jamieson, Bryanna Parkoff

20 April 2022

UNIVERSITY OF THE INCARNATE WORD

This project and the preparation of this report were funded in part by the School of Science, Math, and Engineering Cyber Security Scholars through an agreement with the University of the Incarnate Word.

EXECUTIVE SUMMARY

For our report, the team worked to deploy a static web server hosted through Amazon S3 which would function as both a website host and the backend database. Utilizing the core features of S3, our website acts as a video game database and repository for file downloads. The website, a clone of *Vimm's Lair*, acts as our team's proof of concept. It holds not only the website core files, but also the ISO files to be distributed to the end user all in the same system. AWS S3 provides built-in auto-scaling and bucket security while site user metrics are maintained and displayed through Amazon CloudWatch. By the end of the project, the website was functional and within scope, though there was a minority of ISO links which did not operate properly.

Our Website

HTTtrack

Brackets

TABLE OF CONTENTS

Project Schedule Management

GANTT CHART

Game Kingdom Game Repository (PROJECT NAME) | **1/20/2022** START DATE | **4/20/2022** END DATE | **4/18/2022** LAST UPDATE DATE

Task ID	Task Name	Start Date	End Date
1	Brainstorm	1/24/2022	2/9/2022
2	Plan Cloud Architecture	2/12/2022	2/15/2022
3	New Plan and Restructure	2/24/2022	3/4/2022
4a	Configure Bucket	3/4/2022	3/10/2022
4b	Gather ISOs	3/4/2022	3/10/2022
4c	Clone Website	3/4/2022	3/10/2022
5a	Edit HTML	3/10/2022	3/20/2022
5b	Reconfigure Bucket	3/22/2022	3/24/2020
6a	Test CloudFront	3/24/2022	3/26/2022
6b	Troubleshoot CloudFront	3/26/2022	3/30/2022
7	Test Different Account	4/1/2022	4/10/2022
8	Configure CloudWatch	4/11/2022	4/13/2022
9	Troubleshoot Systems	4/14/2022	4/15/2022
10a	Finish Report and Submit	4/16/2022	4/19/2022
10b	Present Report	4/20/2022	4/20/2022

Please use the following GitHub link if you want to access our project documentation. Thank You.

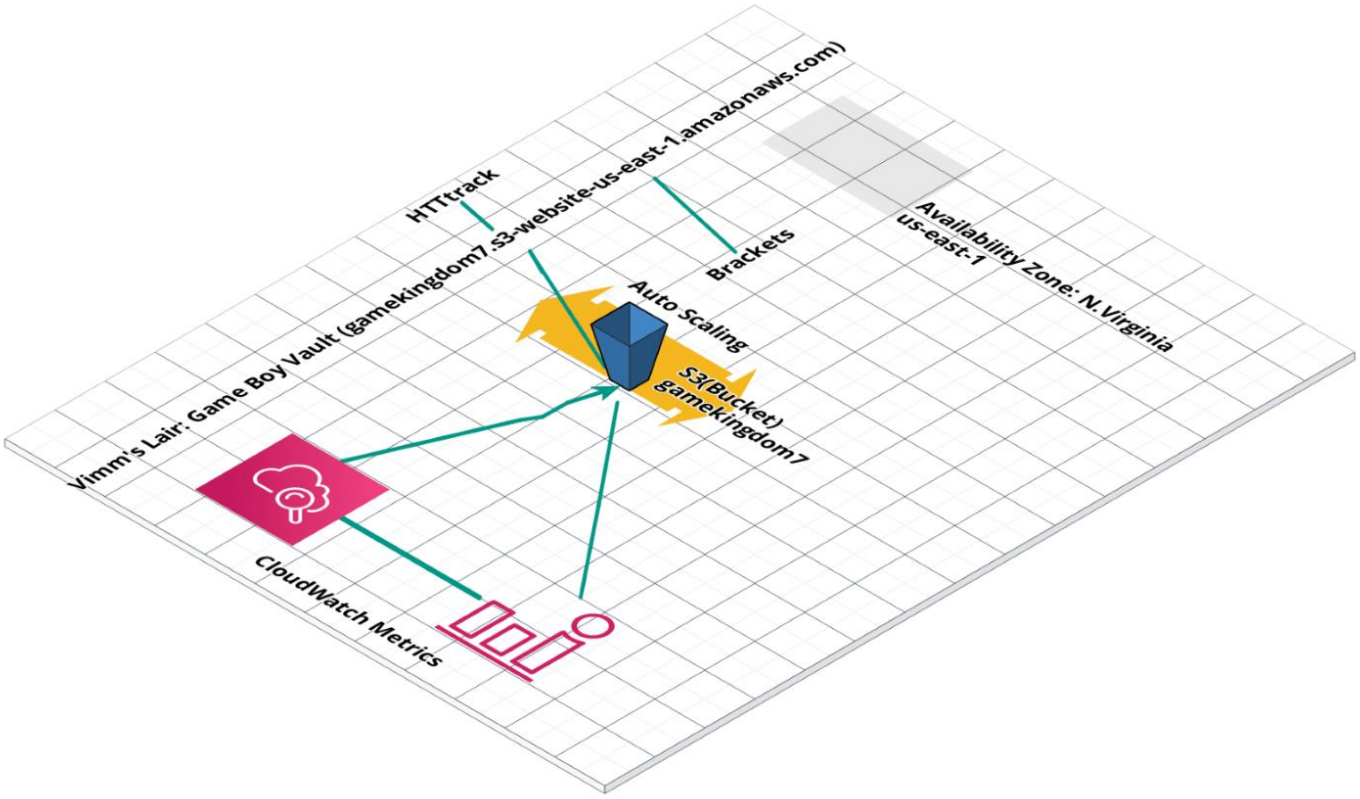
GitHub Repository:

https://github.com/awjamieson83/CloudComputingProject

Milestone 1: Cloud Architecture Set-Up

Our architecture used Amazon S3 (an object hosting service) to host our website and Amazon CloudWatch (a monitoring service) for data analytics [9; 34]. Amazon IAM universal permissions were granted to all project team members for both services. The S3 bucket, gamekingdom7, held both the web pages and the ISOs needed for distribution. The bucket had all public access enabled, static website hosting enabled under properties, and bucket versioning enabled for redundancy. After the website was uploaded and configured (a process described in milestone two below), our S3 bucket was monitored on a CloudWatch widget, which measured the applicable request metrics for our static website on a line graph. This included GET requests, HEAD requests, list requests, bytes downloaded, 4xx errors, 5xx errors, first byte latency, total request latency, and bytes downloaded.

Our security implementation was minimal for this deployment – traditional security groups are not applicable to S3 buckets [2], and public access was permitted since the site was meant to function as a public repository without users having to include sensitive information; further, the bucket's ACL only granted read and write permissions to the bucket owner. The default bucket policy permitted object retrieval.



Infrastructure PaaS Model Description

We used Cloud Craft to model our cloud website architecture, which was a Platform as a Service. We decided to host a static video game website by using AWS S3 Storage. This approach may be considered unorthodox since EC2 grants the ability to host websites through instances. This model of our infrastructure is comprised of S3 Storage (our main service utilized to host all the video game ISO's and web pages from the website Vimm's Lair), integrated scaling (a functionality of S3), Cloud Watch, HTTtrack and Brackets.

S3 Storage is the focal point of our architecture since it is the only truly "necessary" component for the website to function. The website's files and the appropriate ISOs are all stored and hosted on the gamekingdom7 bucket. The site is accessed by visitors through the bucket's URL, which brings the user to the site's main page with a list of ISOs for download.

CloudWatch was used for data analytics; specifically, this service was utilized to track and analyze applicable request metrics for our static website. The nine-request metrics that were tracked were GET requests, HEAD requests, LIST requests, 4xx errors, 5xx errors, bytes downloaded, first byte latency, total request latency, and All requests.

HTTrack and Bracket are third party tools separate from AWS, but they were included in the architecture model due to their importance in constructing our working website. HTTrack was used to clone the web pages for our site from the already existing Vimm.net game repository site. Brackets was used to edit the web pages' HTML code so that clicking on the download button for each game would lead to the endpoint for their respective ISOs in the S3 bucket.

Milestone 2: Deploying the Website

To deploy our website on S3, we began by cloning a page from an existing website. The Gameboy section for the website *Vimm's Lair* was cloned using HTTrack, and after cutting out excess files for the sake of storage and upload time, the necessary web files were entered into the gamekingdom7 bucket. The appropriate ISOs were also entered into the same bucket under the Game_Iso(s) folder. The web pages' HTML code were then edited using Brackets, which allowed us to code the links to download the appropriate ISOs from within the S3 bucket utilizing their specific endpoints. The bucket was then configured under its properties to function as a static website host, using the preconfigured cloned files as the index for our own version of the site.

Amazon S3 takes the deployment from that point, utilizing the files uploaded and pointed to inside the bucket to produce a server that hosts the static site. S3 also manages autoscaling and has access to basic security protocols, such as bucket policies and Access Control Lists (ACLS) [7; 25]. From there, our team verified the proper function of the deployed server through our own testing of links and configured downloads to make sure pages were properly set up and deployed through the directed index file.

Milestone 3: Website Deployment Validation

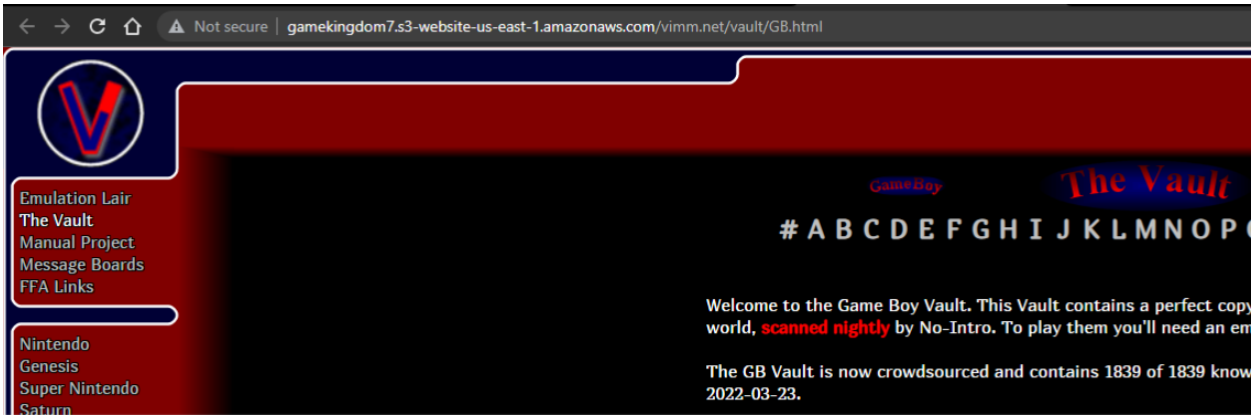
We validated the website performance by visiting the site URL and evaluating each ISO link repeatedly. The majority of links did work, though a few contained errors even after troubleshooting. Of the fifteen unique links, eleven functioned regularly – one gave a 404 error,

and three gave a unique “Error: Database is not responding” message. This issue was validated by several team members before proper troubleshooting began. The process of attempting to fix the links and files started by going back to check the HTML files with defective links; doing so revealed that some HTML files did not have the proper link to the bucket endpoint properly configured, while some others seemed to carry over some error pages through the partial cloning done by HTTrack.

Attempts were made to patch and update the files through manual edits and reuploads, but the edits did not seem to take effect on the website itself. We also manually attempted to navigate to links to check if they were broken on our side, but they had all worked properly up to that point. Our team theorized that a complete reupload may work to bring some of these errors down, but the process would be risky in terms of managing the time and scope of our project at the current point of our project timeline. Our team noted the possible reasons for the broken links and learned from the errors but determined that enough of the site had worked compared to our scope and made the process of a complete reupload risky and out of reach.

Deliverable 1: Hosting Website

As mentioned in milestone three, our website was functional after troubleshooting. Eleven of the fifteen unique ISO links worked consistently, although the remaining four gave consistent errors. The website’s URL is given below:
http://gamekingdom7.s3-website-us-east-1.amazonaws.com/vimm.net/vault/GB.html



Deliverable 2: Auto Scaling

Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost [35]. We would want our static website to be able to oversee large influxes of traffic. Although our website validation did not involve sophisticated traffic testing, we are confident that S3 is more than sufficient to host this site since it is highly scalable and has over 99 percent durability according to Amazon [9; 25].)

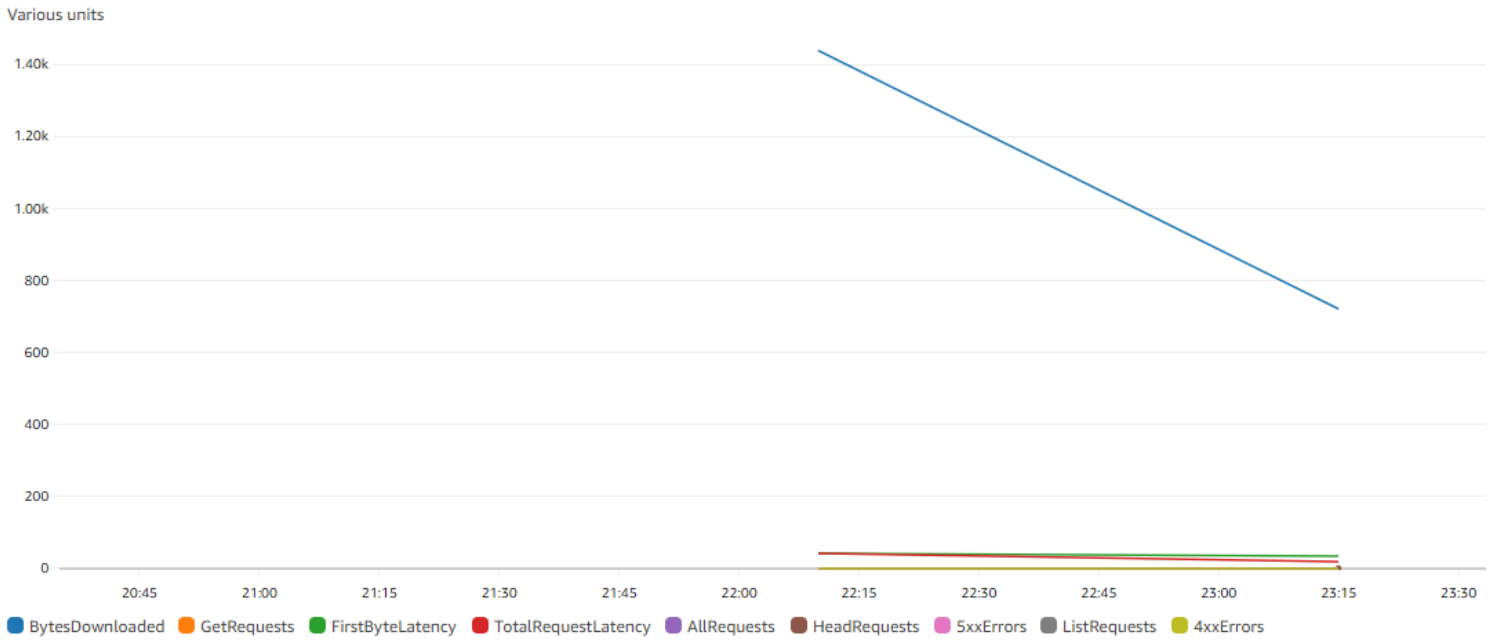
Deliverable 3: Cloud Watch (Data Analytics)

For our third deliverable, we sought to have achieved a reliable method of analyzing the web traffic going to our site. To accomplish this, we utilized CloudWatch. In CloudWatch, there are sixteen request metrics for an S3 bucket. Nine of the sixteen were applicable for our website: Bytes downloaded, GET requests, first byte latency, total request latency, All requests, HEAD requests, 5xx errors, LIST requests, and 4xx errors.

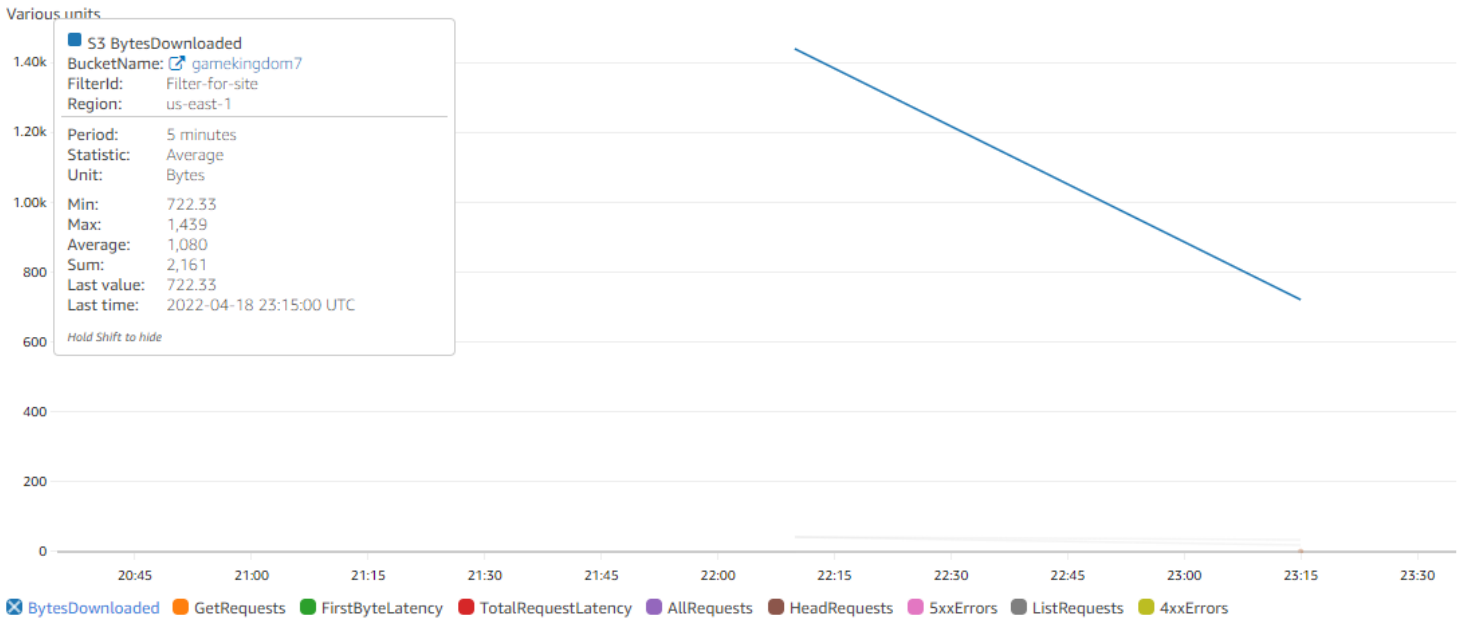
4xx is the family of four hundred errors which cover web pages being inaccessible, due to scenarios such as access restrictions or the material not existing [26]. The HEAD and GET requests are remarkably similar, with both being used to pull objects from a server. The main difference between the two is that HEAD request lacks a response body [27; 36]. First byte latency describes the number of milliseconds until the client receives the first byte of a response from the server [28]. LIST requests are used to return objects to the object [29]. The 5xx family of errors are server-side issues, ranging from traffic overloads to outdated software [30]. Total request latency is the number of milliseconds elapsed between the first and last byte sent on each request [31]. "All requests" is the sum of each applicable request metric for the appropriate time-period. Bytes downloaded refers to the number of bytes downloaded by clients during the appropriate time-period.

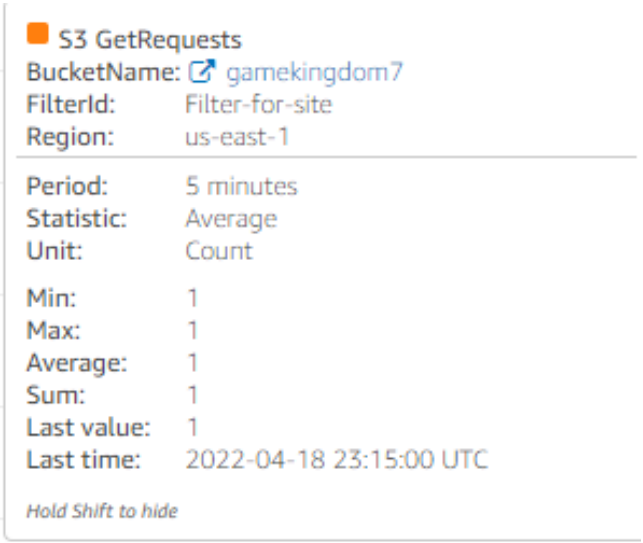
Metric Cards as of 04/17:

Final Results

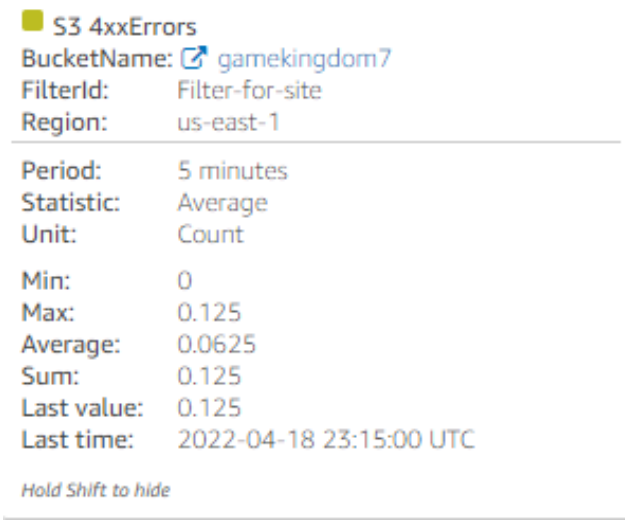


Bytes Downloaded

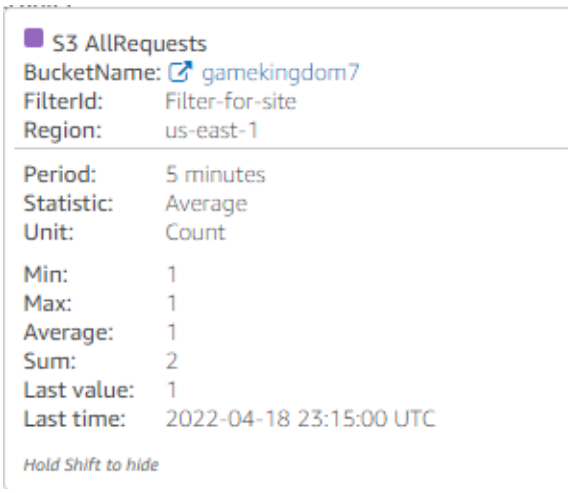




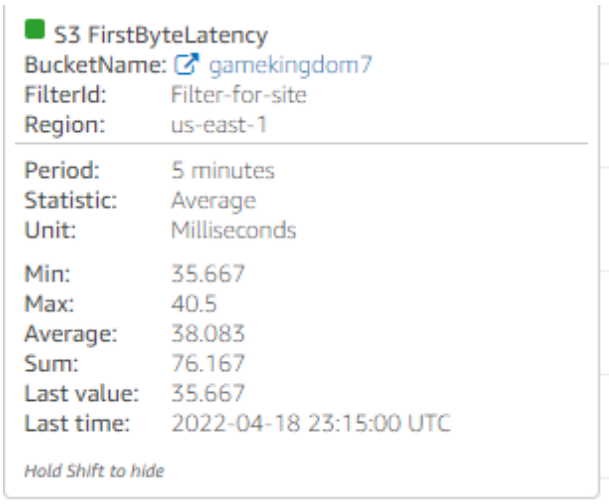
Get Requests



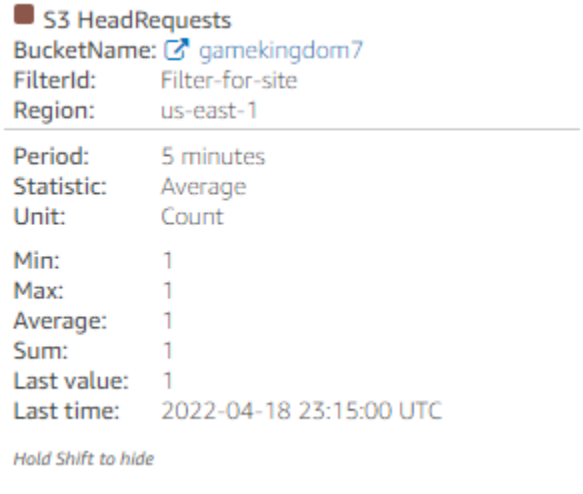
4XX errors



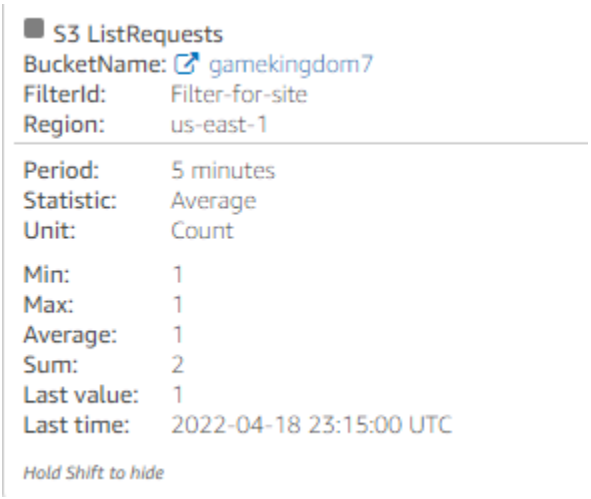
All Requests



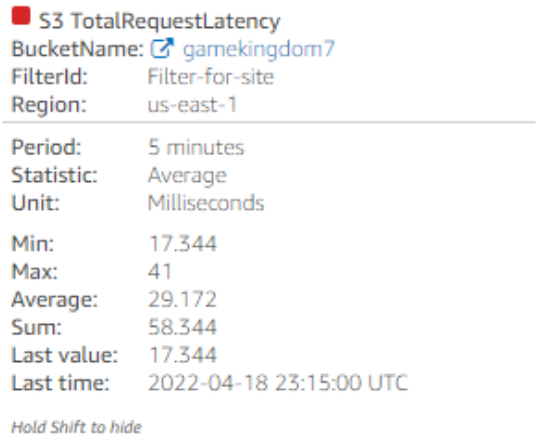
FirstByteLatency



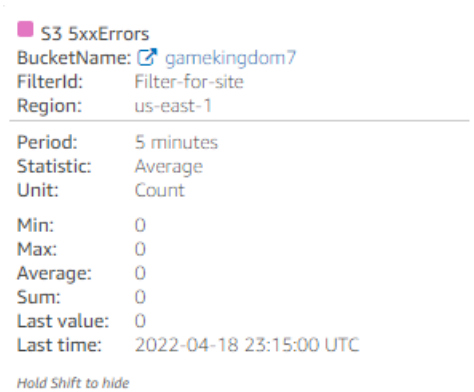
Head Requests



List Requests



TotalRequestLatency



5XX Errors

Growth Reflection

This project has given us a newfound appreciation of AWS's capabilities. Initially, we were having issues deploying our planned website in an EC2 instance, due to issues with configuring Apache and with making AWS CLI operational for all users. S3 provided a stable platform to host our static website without the relative clunkiness of the EC2 instance. Our team learned how to implement this website by researching S3's static site hosting capabilities. We solved this issue by utilizing Brackets to modify the HTML code of our cloned website, *Vimm's Lair*, in order to make the ISO files accessible to site visitors. The team also got around the large file sizes and upload times of the web pages by trimming them after cloning before uploading them to the bucket proper. This helped us learn the importance of utilizing outside tools on cloud architecture deployments.

We also learned how to measure data for our website by using CloudWatch's analytics system to track the appropriate metrics generated for and by our website. Originally, we were going to use CloudFront instead, but this was scrapped in part because it was intended for use by larger companies for global data collection [37].

Letter of Transmittal Sample

April 20,2022

University of The Incarnate Word
Attention: Dr. Gonzalo. D. Parra
4301 Broadway
San Antonio, TX 78209

Dear Dr. Parra:

With this letter, the team Cloud Nation transmits the following items associated with the **CIS 4355 Final Project**.

SCOPE OF WORK (dated 04/20/2022): "Game Kingdom"

DELIVERABLES related to the sub-task:

Aim 1:
Hosting Website
Aim 2:
Auto Scaling
Aim 3:
Cloud Watch (Data Analytics)

Please share these with your team as appropriate. If you have any questions, please contact Bryanna Parkoff at (210)-712-4404 or parkoff@student.uiwtx.edu

Kindest regards,

Bryanna Parkoff
Student of Cloud Computing at the University of the Incarnate Word

Citations Used for Project and PowerPoint:

[1] https://www.tatvasoft.com/blog/cloud-computing-architecture/
[2] https://stackoverflow.com/questions/62869443/how-to-give-full-access-of-s3-bucket-from-ec2-security-group
[3] https://iwrite.sydney.edu.au/ENGINEERING/Reports/Group-Project-Report/Executive-Summary-examples.html
[4] https://sitesy.com/how-to-host-a-dynamic-website-on-aws/
[5] https://Vimm.net/vault/GB/H
[6] https://docs.aws.amazon.com/AmazonS3/latest/userguide/HostingWebsiteOnS3Setup.html#step2-create-bucket-config-as-website
[7] https://stackoverflow.com/questions/60144603/does-s3-auto-scale-by-default
[8] https://austinlasseter.medium.com/how-to-host-a-static-website-on-an-amazon-s3-bucket-be25a613586a
[9] https://aws.amazon.com/pm/serv-s3/?trk=1c5ba169-86e9-4c61-8c90-c507636ad817&sc_channel=ps&sc_campaign=acquisition&sc_medium=ACQ-P|PS-GO|Brand|Desktop|SU|Websites|S3|US|EN|Text&s_kwcid=AL!4422!3!536325599730!p!!g!!amazon%20s3%20web%20hosting&ef_id=CjwKCAjwo8-SBhAlEiwAopc9W_oCy4M8GVEbiJ6AdsmeMA8y2kf6YnwvgTvYxIuU2Si3dIpo-o3C-xoC3HYQAvD_BwE:G:s&s_kwcid=AL!4422!3!536325599730!p!!g!!amazon%20s3%20web%20hosting
[10] https://austinlasseter.medium.com/how-to-host-a-static-website-on-an-amazon-s3-bucket-be25a613586a
[11] https://aws.amazon.com/pm/serv-s3/?trk=fecf68c9-3874-4ae2-a7ed-72b6d19c8034&sc_channel=ps&sc_campaign=acquisition&sc_medium=ACQ-P|PS-GO|Brand|Desktop|SU|Storage|S3|US|EN|Text&s_kwcid=AL!4422!3!488982706713!p!!g!!amazons3&ef_id=CjwKCAjwo8-SBhAlEiwAopc9W68FXbJdBW0vxCnZG8knMqQn47B4Qps4SHpNvz-9bVI0-XiZiF5IyRoCRGAQAvD_BwE:G:s&s_kwcid=AL!4422!3!488982706713!p!!g!!amazons3
[12] https://medium.com/@awsgeek/how-to-get-analytics-from-a-website-hosted-on-amazon-s3-36934454e9d3
[13] https://docs.aws.amazon.com/AmazonS3/latest/userguide/configure-request-metrics-bucket.html
[14] https://havecamerawilltravel.com/how-allow-public-access-amazon-bucket/
[15] https://docs.aws.amazon.com/AmazonS3/latest/userguide/cloudwatch-monitoring.html
[16] https://docs.aws.amazon.com/AmazonS3/latest/userguide/configure-request-metrics-bucket.html
[17] https://www.cloudsavvyit.com/9534/how-to-enable-and-view-request-metrics-for-an-aws-s3-bucket-in-cloudwatch/
[18] https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/publishingMetrics.html
[19] https://www.youtube.com/watch?v=LfBn5Y1X0vE
[20] https://www.techrepublic.com/article/how-to-clone-a-website-with-httrack/

[21] https://www.cloudcraft.co/
[22] How to Get Analytics from a Website Hosted on Amazon S3 | by Jerry Hargrove | Medium
[23] https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lock-overview.html
[24] https://aws.amazon.com/s3/faqs/
[25] https://docs.aws.amazon.com/AmazonS3/latest/userguide/optimizing-performance.html
[26] https://www.bluefrontier.co.uk/company/blog/item/a-guide-to-http-4xx-errors
[27] https://reqbin.com/Article/HttpGet
[28] https://www.searchenginepeople.com/blog/16081-time-to-first-byte-seo.html
[29] https://docs.aws.amazon.com/AmazonS3/latest/API/API_ListObjects.html
[30] https://developer.att.com/video-optimizer/docs/best-practices/5xx-internal-server-error-best-practices
[31] https://www.sumologic.com/blog/monitor-aws-s3-metrics/
[32] https://docs.aws.amazon.com/AmazonS3/latest/userguide/s3-incident-response.html
[33] https://docs.aws.amazon.com/databrew/latest/dg/datasets.multiple-files.html
[34] https://aws.amazon.com/cloudwatch/
[35] https://aws.amazon.com/autoscaling/
[36] https://docs.aws.amazon.com/AmazonS3/latest/API/API_HeadObject.html =
[37] https://aws.amazon.com/cloudfront/
[38] https://iconape.com/brackets-2-logo-icon-svg-png.html
[39] https://usersnap.com/blog/web-application-testing/
[40] https://www.designhill.com/design-blog/essential-graphic-design-tips-for-every-website/