Migrating Knowledge Test of a Driving School to the Cloud

 Andrew Cook

atcook68@cityuniversity.edu

Anil Erturk

erturkanil@cityuniversity.edu

Scott Zhou

zhouscott@cityuniversity.edu

Yu-Che Liu

liuyuche@cityuniversity.edu

Team 5, CS519 Cloud Computing Overview

Master of Science in Computer Science

City University of Seattle

Seattle, 98121, USA

**Abstract** 

Cloud Computing is the on-demand availability of computing resources, which includes computing power and database storage. AWS (Amazon Web Services) is the leading industrial cloud computing provider. The availability of AWS enables companies to deploy their application globally in minutes (AWS Advantages, n.d) compared to the traditional way of deploying their application which is owning and setting up their own server. To help with migrating to the cloud, AWS provides a best practice called a Well-architected framework, which includes operational excellence, security, reliability, performance efficiency, cost optimization.

The organization has many students and employees and has been successfully running face-to-face tests. This paper is a proposal for the company to move all exams online using AWS. Both AWS and on-premises servers would enable them to test remotely. The paper explains why cloud computing is a better option and proposes a plan that can be implemented.

**Keywords:** AWS,Cloud computing, Operational excellence, Security, Reliability, Performance efficiency, Cost optimization

**Introduction**

The company in question is called Driver Education Services. It is a small driver education company that has a small number of students per day and simultaneous students. The company processes a maximum of few students at a time. Because of the pandemic, the company wants to do their knowledge exam online. The paper talks about why the cloud solution is the best choice for this use case as opposed to an on-premises solution. The possibility of the company taking advantage of the AWS free tier since they don’t require many resources makes this solution even cheaper on top of the already cheaper cloud solution. Other advantages such as the decreased need for networking infrastructure and employees are also discussed. The five pillars of AWS cover all the advantages of the cloud and give a framework to build high-quality cloud solutions that can deliver on the requirements.

1. **HOW THE FIVE PILLARS OF AWS HELP DELIVER ON EXPECTATIONS**

Jumping directly into a software project without spending time on requirements, environment, and design increases the probability of wasting time developing a system that will not work. Software projects that include on-premises or cloud servers have the same problem. Therefore, it is necessary to first analyze the problem, list the requirements, choose the environments and services, and work on the design. Having a well-thought-out architecture that incorporates the 5 pillars of AWS will help produce stable and efficient systems. The 5 pillars help the company deal with non-functional requirements and allow the company to focus on functional requirements.

When architecting technology solutions on AWS, if you neglect the five pillars of operational excellence, security, reliability, performance efficiency, and cost optimization, it can become challenging to build a system that delivers on your expectations and requirements (Belt, 2018).

These five pillars will help the customer understand why a cloud is better than on-premises and show how nonfunctional requirements are handled with the cloud.

**Operational Excellence**

In achieving the business objectives, the ability to run effective workloads and continuously improve procedures and processes is of the utmost importance. Operational excellence is accomplished by incorporating the best principles and practices that will help achieve the business objectives in the cloud. Toward this goal, 5 design principles exist to support processes and build value for the company. The first of these is performing operations as code (Carlson et al, 2020).

To improve company performance, workload operations can be automated. In fact, your entire cloud environment can be automated and updated in the AWS script, mechanizing the response execution, reducing human error, and keeping consistent responses (Carlson et al, 2020). This means that instead of time-consuming graphical user interfaces, automating these operations in the AWS script will yield great benefits. Making frequent, small, reversible changes is the second design principle and offers the benefit of reversing the change if it doesn’t solve the issues. The idea here is to design each workload to have regular, incremental updates. In the effort for continuous improvement, creating space and time to review procedures is of great importance. Being able to and having scheduled time to review standard operating procedures (SOP’s) to validate their effectiveness and efficiency is another design principle (Carlson et al, 2020).

Anticipating failure is also a design principle. It is important to test the response procedures to determine their sufficiency to identify sources of failure. Schedule game days to test the organization's response to simulations. This is important because it creates familiarity with the procedure's execution. The last design principle is learning from all operational failures. Capturing the lessons learned is a cornerstone of process improvement. This information is needed by the rest of the organization to not repeat past mistakes (Carlson et al, 2020).

Serverless architecture is an important contributor to operational excellence. AWS Lambda provides this with its’ event-driven, serverless compute service. The central idea behind going serverless in the way of operational excellence is continuous improvement in processes and procedures. The main benefits of applying Lambda, in this case, begin with no servers to manage. This means that Lambda will run code without the need to provision infrastructure. Another advantage of the serverless power of Lambda is the scaling potential. AWS will automatically run code in reaction to events. Your code will run in parallel and scale to the workload, no matter the size. In addition to the continuous scaling, the customer only pays for consumed compute time. Only those milliseconds that your code is run and the number of times the code is triggered are the items that AWS charges for (Thakkar, Brooker, Surkatty, 2021).

In addition to the above direct benefits, there are several indirect benefits that bring advantages along with. These indirect benefits are the increased innovation, the AWS ecosystem, and the AI-upgraded applications. AWS has a serverless application repository for publishing applications with various integrated development environments like the AWS toolkit for Visual Studio and AWS Cloud9. The tools in this ecosystem enable the company cloud architects to build applications that can take advantage of these capabilities. In addition to all this, the lack of attention on management of the server infrastructure means the programming resources can refocus their attention on innovation and development. Therefore, it is extremely important to apply AWS Lambda to the serverless architecture of the company's web presence (Thakkar, Brooker, Surkatty, 2021).

**Security**

If committed to the serverless approach to company performance, there is one area that is of critical importance, security. The pillars of security include identity and access management (IAM), the ability to protect assets, systems, data, as well as to utilize cloud technology toward improving your security. These principles are applied using critical design principles to aid companies in strengthening workload security. These are enabling traceability, applying security at all layers, automating security best practices, protecting data in transit and at rest, keeping people away from data, implementing a strong identity foundation, and preparing for security events (Potter et al, 2020).

The best place to start with building a secure cloud presence is with best practices. Following the security pillars’ design principles, the first area is IAM. Assuming the company will need to start with multiple users and separate groups, the best practice is to use a centralized identity provider. AWS Single Sign-on meets this requirement by becoming the specific location, or default zone, from which to grant users access to accounts (Potter et al, 2020).

Detection is an incredibly significant part of the cloud security architecture and as such, a foundation on which the design principles are based. There are three distinct areas of detection, configuration, and investigation. Configuration deals with the establishment of detection mechanisms. AWS CloudTrail is a great tool for providing event history of account activity, keeping a log of AWS management console activity, command-line tools, and others. AWS Config is another service that records data. This tool monitors the configuration of the currently equipped resources and evaluates them against the specified set. A cornerstone of security is Amazon GuardDuty, which provides a malicious activity detection service. Unauthorized activity is also monitored and recorded (Potter et al, 2020).

The investigation is the other area of approach and foundation of the design principles. It is recommended that a playbook (or process to investigate), accompanies each detection mechanism that is enabled. With this, each time a trojan is detected, a quick investigation and remediation process is followed. A good tactic here is to equip Amazon EventBridge to utilize CloudTrail activity history and build a scalable rules engine. In addition to EventBridge, AWS Config is a service that detects alterations to in-scope services and helps with the enforcement of compliance (Potter et al, 2020). While these tools can be extremely effective in providing security for the cloud presence, they are only tools and must be researched and equipped to secure the cloud presence.

**Reliability**

Resiliency is the ability of a system or infrastructure to respond to service disruptions or dynamically scale up the computing recourses based on the demand.

In Amazon Web Service, it provides five different availability categories: 99%, 99.9%, 99.95%, 99.99%, 99.999% available. Each category provides a different level of downtime and recovery time. In this project, this paper selected to achieve 99.999% availability, which provides the system or application without almost no downtime, and the recovery time will be under 12 minutes. (AWS, 2020. AWS Well-Architected Framework.)

To achieve this high availability, this project will use the Warm Standby approach across two Regions. Each resource deploy in each Region needs to be monitored. For instance, two servers are running in east-1 Region. When the server goes down, the monitor service will trigger a failover procedure to switch to redirect the traffic to another Region. The monitor service can aggregate all the logs into the S3 bucket for a future business analysis on every problem. Amazon RDS enables a Multi-AZ solution to create one database for primary use, and the other database will be in standby mode. (AWS, 2021. AWS Reliability Pillar Well-Architected Framework.)

Changing the infrastructure can take up to several months to let the machine ready to use. It is important to have flexibility on the infrastructure. To achieve this, Elastic Load Balancing is implemented to auto-scaling the website and the application. When the application is under downtime or too much traffic, Elastic Load Balancing can trigger an autoscaling feature to scale out the application to increase the availability. The database can also adapt to change in demand. Our application used Aurora RDS as a database to store users’ information. Enable autoscaling storage and deploy read replicas in multiple availability zones to increase the availability. (AWS, 2021. AWS Reliability Pillar Well-Architected Framework.)

Updating the software and dealing with undetected problems can affect the system’s reliability. For this paper, the Blue/green deployment method is selected to deploy the infrastructure. Using this Blue/green deployment method, the legacy system and new production system will exist simultaneously. If an unexpected problem happens in the new production, the system administrator can use Amazon Route 53 to redirect the user’s traffic to the legacy system. (AWS, 2021. AWS Reliability Pillar Well-Architected Framework.)

Back-up data is one of the key elements in availability. When the database or EBS volumes become inaccessible, the system or website cannot run properly. To minimum the data loss, automatically configuring backups on the database is important. For this paper, the solution for this project will use Amazon Data Lifecycle Manager to manage EBS snapshots and enable Point in Time Recovery for the Amazon RDS. (AWS, 2021. AWS Reliability Pillar Well-Architected Framework.)

**Performance Efficiency**

AWS summarizes performance efficiency as “The ability to use computing resources efficiently to meet system requirements and to maintain that efficiency as demand changes and technologies evolve.” (Belt, 2018). Getting work done is relatively easy when efficiency is not considered. Though this is wasteful and expensive. Engineering is about getting the work done and getting it done efficiently. Servers are inherently prone to either not having enough resources or not using all their resources. So, when it comes to servers, having a shared ecosystem with a big pool is a good idea because if Company A requires more resources, they can request that from the big pool and if Company B has more resources than they need, they can release extra resources into the big pool. Since this adjustment can be done automatically and easily, it can be done continuously over time as the resource demands of users fluctuate.

AWS gives small companies like drive-ed access to the same technologies as everyone else. The democratization of technology refers to technology becoming more accessible, and AWS has been opening more and more regions, availability, zones, and edge locations (AWS Regions, n.d). AWS has also been working on making more and more services available in more regions. (AWS Regional Services, n.d).

AWS also allows anyone to set up a server within minutes and even go global within minutes (AWS Advantages, n.d). The fact that AWS has been opening more regions also makes this point stronger. In order to go global, deploy close to end-users, the customers of AWS require regions, availability zones, and edge locations. This also means that there are no performance, bandwidth, and latency trade-offs with the cloud.

AWS cloud also allows for experimenting that companies couldn’t do in real life with physical servers.  Cloud offers many choices for every service. It allows customers to try new configurations with the click of a button. And if the customer isn't satisfied, they can simply cancel the service and try another service.

Large initial cost and lack of server infrastructure are big problems for companies trying to set up servers. Cloud offers everyone easy and cheap access to servers.

Lack of employees who are skilled in servers is another big problem for companies trying to set up servers. Cloud takes responsibility for many layers of servers, thus decreasing the need for employees who are skilled in those layers of servers. Cloud services most commonly take over responsibility for infrastructure, but this arrangement changes from service to service.

**Cost optimization**

Cost optimization is an important feature to keep business expenses with the budget. AWS allows the user to control the cost and continue to drive spending and cost reduction while maximizing business value. AWS offers various services and pricing options to meet the required performance and capacity while maintaining the budget. Especially during this period of crisis, AWS will work with the organization to help achieve its financial goal.

Cloud Financial Management is a good starting point to build capability via knowledge building, programs, resources, and processes. The goal is to ensure financial success and quickly implement business value realization in the cloud. Pay only the resource that is required will make it easy to change the usage upon on business requirements. For example, the development and test services usually happen during weekdays. Stop these services while not using them could potentially save up to 75% of percent cost. Furthermore, AWS offers free tiers in many services so users can test the water without any cost before they move to the cloud. AWS offers a measurement tool to evaluate the workload and its costs so the organization can track the gains from increasing output and reducing costs. AWS will tremendously be saving the cost for heavy lifting of data center operations such as racking, stacking, and powering servers. Also, eliminate the operational burden for managing operating systems and applications which results in more focus on the business it-self than IT infrastructure. The user can monitor the usage and cost of systems that could be used to measure return on investment and further optimize the resources and costs.

Migrating into the cloud will make projects laugh faster because of shortened approval, procurement, and infrastructure deployment cycles. Every organization has different units associated with different properties. AWS allows the user to align their financial objects and achieve them which results in improved efficiency. AWS has Cost Explorer to give analysis for cost and usage awareness throughout the organization. AWS Budgets can show proactive notifications, and AWS blogs give information on new services and features.

AWS help manages demand and supply resources. You only pay for what you need. The user provision how many resources are needed at a specific time, and not paying for these services when not using them. Users can also change the demand with throttle, buffer, or queue to serve with less resource which lowers the costs. Users can make AWS automatically provision the resources for its workload demand. Auto Scaling adds or removes resources as needed. Amazon API Gateway and Amazon SQS are good examples to implement throttling or queue in the workload.

AWS has regularly updated its services and features. It is highly recommending the user review their existing architecture in order to keep it cost-effective. Decommissioning resources, entire services, and systems as the requirements change. Developing new resources or features can optimize the workload. This ensures the services stay updated and operating efficiently while keeping the low cost. The user can even add or replace a component with its new service to help increases inefficiency. It is crucial that businesses regularly review and change their services. (AWS Cost Optimization, n.d) (Potter et al, 2020)

Serverless computing lower the cost by utilizing a pay-for-value billing model, the user won’t need to pay for over-provisioning and could easily optimize resource allocation in order to manage the budget.

AWS Lambda only charges for the actual usage of compute time and resources, so there are only things that need to focus on in order to optimize the cost. Lambda compute time will be based on primary three things: resources allocated, language runtime, and the code itself. To figure out the right optimal memory size, users need to test the Lambda function at different resource levels to ensure the optimal level of price is for the application. It is an anti-pattern to think by using the smallest resource size will offer the lowest total cost. The smaller resource size could cause longer execution time which results in more cost. It will be best to have more available resources to run the function faster. AWS Lambda can make the application run asynchronous. Each component from the application architecture could be executed separately instead of bond together and awaiting responses to synchronous requests. Asynchronous design is fit well with AWS Lambda and is helpful to application module and decouple functions which result in a more cost-effective manner. Some Lambda event sources allow users to configure the batch size of the number of records passed on each function such as DynamoDB. The user needs to test to find out the optimal batch size so each event resource is adjusted to how fast the function can finish its task. Lambda has many event resources available to offer various solutions to meet the application requirements (request scale, the volume of data, latency required, etc.). These event sources may affect the total cost of architecture due to which AWS service was selected as a component for the Lambda function.

Serverless computing lowers the cost by utilizing a pay-for-value billing model, the user won’t need to pay for over-provisioning and could easily optimize resource allocation in order to manage the budget. Serverless cost is not only related to the level of usage, but also the architectural decision that affects cost efficiency. These will become more important as the traffic grows, so different designs and patterns are also a key factor of cost-effectiveness.

There are five common areas that could save money when designing a web application: API management layer, data layer, content distribution layer, integration layer, business logic layer.

API connects the frontend alien and the backend in serverless web applications. AWS API Gateway is the most common choice because of the fully managed service and scales automatically. AWS API Gateway offers HTTP APIs which cost 70% less but still has many features in the REST APIs. It also supports Lambda service integration and simplifies the deployment model than REST APIs.

Amazon CloudFront is a content delivery network that enables the user to distribute the content globally without worry about any infrastructure. This service reduces the latency when users need to access from different geography and reduces the load on other parts of the service. CDN has a built-in distribution of application feature which help reduce the builds become too larger than necessary. Another use case is optimizing user-generated content. Many users upload images to the apps so they can share them with others, but high-resolution images slow down performance and cost more for data transfer. Resizer Lambda function can optimize uploaded assets into 98-99% smaller to make a significant difference for a busy web application.

AWS offers a variety of database and storage solutions for web applications. Billing models is based on the region and service. Data access and storage requirements of the application will determine to make decisions for which service to use. Store binary in S3 is more cost-effective than a database because S3 uses preassigned URLs when the user uploaded the data instead of proxying data with other AWS services. Amazon DynamoDB is also a great choice due to the pricing model is based upon usage and storage.

Amazon SQS, Amazon SNS, and Amazon EventBridge are essential to decoupling serverless applications. These services using a request-based pricing component that 64 KB of a payload will be charged as one request. There are two methods used for optimization in web applications: combine messages and filter messages. Combine multiple messages lower the total number of publishing actions to SNS, also removed unused attributes in the message so the user can have more data in a single request. Filtering out non-useful messages can reduce the compute time which results in lower cost. SNS can eliminate messages from the content base on some criteria. For complex filtering options, EventBridge will be a better choice because it offers more filters criteria such as prefix matching, numeric bathing, and other patterns, combine these rules into a single filter.

Step functions allow users to combine AWS services into a serverless workflow in a short amount of time. Standard Workflow is billed per state transition while Express Workflow is billed on requests and duration. Express Workflow is more cost-effective when dealing with a large number of events in a short duration. Compare these two types of workflow and switch across based on the application requirements.

The pay-per-value model fits well for web application backends. Design choices and service configurations will become more important as traffic grows in order to optimize the cost.

**2. MOVING TO THE CLOUD**

**2.1 Cloud Concepts**

**Cloud Service Models**

The company requires a Platform as a Service (PaaS). Infrastructure as a Service (IaaS) is not necessary since the company doesn’t want or need to handle the hardware itself. Software as a Service (SaaS) is again not needed since the company wants to use its own web application.

**Cloud Computing Deployment Models**

The company doesn’t require or interested in a private cloud or hybrid cloud deployment. Therefore, the company will use a pure cloud deployment.

**Technical Support Models**

AWS offers four support plans. Since this company will not be doing too many and complicated developments in the cloud, there is no need for the company to spend extra money on Developer, Business, and Enterprise level supports. The company can use the free Basic support that include Resource Center access, Service Health Dashboard, product FAQs, discussion forums, and support for health checks.

The company can also access the AWS Pricing Calculator and the Billing dashboard for free in order to learn about prices before purchasing services and keep track of their past, current, and future monthly costs.

**2.2 AWS Global Infrastructure**

Using the region that is closest to the company's users will reduce the latency for users and the company's systems. Websites like Cloud Ping can be used to determine latency to users. In this case, us-west2 Oregon is the best region with 29ms latency (Cloud Ping, n.d). Since the company is local and not global, for now, they will not need to use multiple regions (AWS Regions, n.d).

**Amazon CloudFront**

The company’s target crowd is people who want to have the online driving test, which those people may not stay in the same area. Some of them may use their time to practice driving test questions during vacation or business trips. Latency becomes an important issue. Amazon CloudFront can provide different vary edge locations for the application cache. Applications source code will be distributed to the edge locations for providing low latency connectivity. (Amazon Cloudfront 2012)

**2.3 AWS Cloud Security**

**Securing the New AWS Account**

To begin the journey of securing the cloud presence, the company should save the account keys if needed. Groups should be created, roles attached to groups, and users created and assigned to groups. Having a centralized identity provider is another critical best practice to follow. Based on the assumption that multiple users and groups will be created, AWS Single-Sign-On (SSO) accomplishes this objective by being the location which grants access (Potter et al, 2020).

**Multi-Factor Authentication**

Multi-Factor Authentication (MFA) should also be enabled for extra security of the IAM accounts. It’s a very simple practice which adds an extra layer of security in addition to the username and password. This can be enforced with software or hardware mechanisms. Enabling an “always-on” or “context-aware” configuration in tandem with SSO as the identity source (Potter et al, 2020).

**AWS CloudTrail**

AWS CloudTrail will allow the company to monitor activity on their account. Providing event history, continuously, and retaining this history. Making internal compliance as well as meeting external regulations much easier. This is in addition to tracking of changes and detecting unusual activity with CloudTrail Insights (Amazon Web Services, 2014).

**2.4 AWS Networking**

**Route 53**

The company has two main services. The static website (drive-ed, n.d) and they want to create a dynamic web application for their exams. The company doesn’t want to use generic AWS domains and subdomains therefore they will continue using their own domain for the static website and they can have the dynamic web application either in a different subdomain or in the same subdomain and domain but in a different path such as http://www.drive-ed.com/ exam. The dynamic web application can be programmed so, that path corresponds to a Java servlet or some other dynamic web application method. The company will want to use Simple routing for their single-server environment.

**Elastic Load Balancing**

Elastic Load Balancing distributes traffic across multiple resources. For this static website, application load balancing is used for distributing the traffic. When our S3 hosted static website goes down, it can trigger Application load balancing based on failover policy to direct most of the traffic to another s3 bucket, which contains the same static website. The key step for implementing application load balancing is that Route53 should point to the application load balancing, and the application can base on the healthy check inside each s3 bucket to decide which S3 bucket website is available to the public. (Configure route 53 failover from the elastic load balancer to an S3 hosted static website. 2014)

**2.5 AWS Compute**

**AWS Lambda**

The static webpage won't be accessed often since this is a small company. The exam dynamic webpage will be accessed only when a student is taking the exam. This means that this company is a very good candidate for the serverless compute service AWS Lambda. This prevents wasting resources and decreases costs.

**2.6 AWS Storage**

**AWS S3**

The company can use cheap S3 storage to store the static website. S3 one zone infrequent access will be very cheap since it doesn’t offer as much reliability and is not as fast as the S3 standard. The contents of the static website should be backed up outside of the cloud by the company in case of a disaster though.

**2.7 AWS Database**  
  
**AWS RDS**

AWS RDS takes away some of the administrative work from our hands. This allows the company to focus on our data. It offers six database engines such as the open-source MySQL. The database can be created in a private VPC in order to isolate it from the internet. RDS has an automatic synchronous copy in another availability zone in the same region. It even allows for read replicas of databases. Reserved instances make more sense in this use case since the company will need the database for long term. But depending on the frequency and duration of database access, using on-demand instances when they are needed, and exporting the data out to a text file in the S3 bucket when the database is not needed could also be a solution.

**REFERENCES**

Amazon Cloudfront. (2012).   
https://aws.amazon.com/cloudfront/

Amazon Web Services. (2014, November 10). AWS CloudTrail User Guide (Rep.). http://awsdocs.s3.amazonaws.com/awscloudtrail/latest/awscloudtrail-ug.pdf

AWS Advantages. (n.d). Six Advantages of Cloud Computing.

https://docs.aws.amazon.com/whitepapers/latest/aws-overview/six-advantages-of-cloud-computing.html

Belt, D. (2018). The 5 Pillars of the AWS Well-Architected Framework

https://aws.amazon.com/blogs/apn/the-5-pillars-of-the-aws-well-architected-framework/

drive-ed. (n.d). Driver Education Services.

http://www.drive-ed.com/

AWS Cost Optimization. (n.d.). Cost Optimization. re  
 https://wa.aws.amazon.com/wat.pillar.costOptimization.en.html

AWS. (n.d.-a). Serverless Architectures with AWS Lambda.

<https://docs.aws.amazon.com/whitepapers/latest/serverless-architectures-lambda/cost-optimization-best-practices.html>

AWS. (n.d.-b). Serverless on AWS Build and run applications without thinking about servers.

<https://aws.amazon.com/serverless/>

Beswick, J. (2020, October 19). Optimizing the cost of serverless web applications. AWS Compute Blog.

<https://aws.amazon.com/blogs/compute/optimizing-the-cost-of-serverless-web-applications/>

AWS Regions. (n.d). Regions and Availability Zones   
https://aws.amazon.com/about-aws/global-infrastructure/regions\_az

Cloud Ping. (n.d). Latencies to AWS Regions

https://www.cloudping.info

Configure route 53 failover from elastic load balancer to an S3 hosted static website. (2014, March 18).   
https://technologysnippets.wordpress.com/2014/03/18/configure-route-53-failover-from-elastic-load-balancer-to-an-s3-hosted-static-website/

AWS Reliability Pillar AWS Well-Architected Framework. (n.d.).   
https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/wellarchitected-reliability-pillar.pdf#reliability

AWS Regional Services. (n.d). AWS Regional Services

https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/

Carlson, B., Steele, J., King, R., & Mitzsimons, P. (2020, July). A Well Architected Framework: Operational Excellence Pillar (Rep.).

https://docs.aws.amazon.com/wellarchitected/latest/operational-excellence-pillar/wella nn, B., Potter, B., Shinn Johnson, B., Pogson, B., Boyd, D., & Walker, D. (2020). AWS Well Architected Framework: Security Pillar (Rep.).

https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/wellarchitected-security-pillar.pdf#welcome rchitected-operational-excellence-pillar.pdf#welcome

Potter, B., Shinn, B., Johnson, B., Pogson, B., Boyd, D., & Walker, D. (2020, July). *AWS Well-Architected Framework: Security Pillar* (Rep.). https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/wellarchitected-security-pillar.pdf#welcome

Thakkar, M., Brooker, M., &amp; Surkatty, O. (2021, February 15). Security Overviewof AWS Lambda.   
https://docs.aws.amazon.com/whitepapers/latest/security-overview-aws-lambda/security-overview-aws-lambda.pdf#security-overview-aws-lambda