# Cloud Computing: The Benefits of Migration

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

**Nick Teller**

Fall, 2023

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Briana Morrison, Department of Computer Science

Rosanne Vrugtman, Department of Computer Science

# Cloud Computing: The Benefits of Migration

CS4991 Capstone Report, 2023

Nick Teller
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
nst6tx@virginia.edu

**ABSTRACT**
The company I interned at over the summer needed to improve operational efficiency and save costs on their internal application and databases. To accomplish these goals, I aided in the migration of software to the cloud. Utilizing some of the many services available on Amazon Web Services (AWS), I was able to transition some critical components from on-premises datacenters to the cloud. The migration will have many benefits, including updating legacy programs, simplifying debugging, and adding new functionality. The full switch to the cloud still has several steps to go, and more applications need to be moved and tested working together as a whole.

## 1. INTRODUCTION

When I first joined my team, I learned that my team was assigned to help with a company-wide cloud migration. I was not expecting this, and I knew I had to learn how to work with AWS quickly.

Over the course of my 10-week internship I gained lots of hands-on experience with software development in the cloud. There was a lot to learn, as there are AWS provides many services**,** to use, each of which can be integrated with both internal and external applications. Additionally, I had to know when one service would be better to use than a similar one. With this knowledge, I could help my team transition important applications to AWS.

## 2. RELATED WORKS

One of the most helpful resources I found for learning about AWS services was an introductory online video course from O'Reilly Media [1]. These videos provide an overview and basic tutorial on the essential cloud services offered by AWS. Furthermore, I found that gaining additional information about which types of businesses benefit from a cloud migration and how they benefit was useful.

After doing some background research into this topic, the most insightful source was a by Abdoulaye [2]. More specifically, the chapter titled “The Digital Economy's Challenges, Opportunities, and Relevance of AWS” details the potential issues and rewards of a cloud migration, which applies to the company I interned at and most other large businesses.

## 3. PROJECT DESIGN

This section contains most of the details on my work this summer. Subsection 3.1 is about my onboarding and learning about what technologies my team was working with. Sections 3.3 and 3.4 cover my specific assignments in greater depth.

### 3.1 Onboarding

My team used a lot of terminology and software I was unfamiliar with, and for the first four weeks I was mainly educating myself on the concepts and skills I would need to use for the rest of the internship. There was extensive internal documentation about the proposed cloud migration design, which allowed me to familiarize myself with what the team was trying to accomplish. These documents showed which AWS services were going to be used for each step in the transition and how these services worked together.

The main services I would be working with were Simple Queue Service (SQS), Simple Storage Service (S3), AWS Lambda, AWS CloudWatch and AWS CloudFormation (CF). SQS queues messages for other software components to process, S3 allows for the buckets to store in the cloud, Lambda allows users to make functions triggered by events, CloudWatch is a logging service, and CF templates allow for cloud infrastructure to be made from code [1]. The online video course from O'Reilly was incredibly helpful in understanding these and other services, and the official AWS documentation available online was also useful. During this time, I also learned how to use the different developer environments with Jenkins and how to use Bitbucket for version control.

My final step of this onboarding period was updating old information in the current build of the application. This was a simple but necessary task that would test whether I was capable of implementing what I had been studying and ready for more complicated assignments. It took me a bit to get used to actually working with Jenkins and Bitbucket but I was able to accomplish this goal without any notable issues and progress to helping my team with the cloud migration.

**3.2 Multistep Report Generation**

My first story (a term for part of the project) was moving a multi-part job to the cloud. This job generated a report based on internal data, but each part of this report could only be created after the previous part was finished. This process was vital to the application usefulness and needed to be migrated to AWS. To implement this I worked with another team member.

Whenever a report needed to be made, a starter message would be sent to a SQS queue. The code we wrote would listen to the queue, and when this initial message was present it would trigger the first part of the report generation. When step one finished, it would send a new message to the queue stating part two was ready to start. This process of queueing a new message, listening for it, and starting the next part of the report repeated until the whole report was finished and could be used internally.

Coding the described implantation had its fair share of challenges, and the logs from CloudWatch provided were the only way to tell if something was working properly or not. Nevertheless, we were able to get this procedure to work and made progress in the cloud migration.

**3.3 S3 File Copy**

My team's application stored many files and objects in S3 buckets in the early parts of the transition to AWS. However, some buckets needed to access certain data stored in other buckets, and fixing this problem with a Lambda function was my next assignment. Additionally, this Lambda function would need to trigger every time a file was added to a folder in one of the S3 buckets.

My first step was actually coding this function and making sure it only accessed folders within the buckets for which it had permissions. In order for me to deploy my

function in a test environment, I had to use a CF template. These templates can be very finicky, and proved to be the biggest hurdle I overcame during my internship. After a lot of trial and error with syntax, I was finally able to get the template to properly deploy the Lambda function and properly test it. On the last day of my internship, I completed this story and was able to leave my team with one last contribution.

## 4. ANTICPATED RESULTS

My work this summer helped my team move a critical internal application to the cloud. As of the end of my internship, this migration was not yet finished but substantial progress had been made. First, hosting applications on-premises is more expensive than on AWS, so transitioning core internal software saved money. Second, my team's work moved the application to running on modern infrastructure. While working on the multi-step report generation, I found that the original code was from the early 2000s. This may not seem like that long ago, but software has come a long way since then. This evolution also applies to debugging. Whenever I had to use debugging services other than CloudWatch, it took me significantly longer to learn how they worked and, therefore, to solve bugs.

Last, utilizing AWS makes it far easier to add new functionality to applications. AWS has hundreds of services, many of which are designed to work in tandem. When relying on multiple third-party programs to run an application on-premises it can be hard to add a completely new feature when another team requests it. But when working with AWS, there is lots of documentation and tutorials on how each service works and how to use it with compatible services.

## 5. CONCLUSION

Cloud computing is a very prominent area in the computer science world right now, and many companies are undergoing or considering a migration. It is often cheaper, simpler, and more efficient to use cloud services instead of traditional on-premises data centers. Having skills and experience with AWS is great for an aspiring software developer, and I was able to take full advantage of this opportunity.

## 6. FUTURE WORK

Although a lot of work was completed during the summer, there is still much to be done. The full functionality of this internal app needs to be migrated to AWS, which will require planning the services to use, creating that feature in the cloud, and making sure it works with all of the previous work. Eventually the full application will be transitioned, and it will be fully operational in the cloud. New functionality, bug fixes, and quality of life improvements will happen eventually, as the app still needs to be supported post migration.

## REFERENCES

[1] Smith, C. (2022). Amazon Web Services (AWS), 3rd Edition. O'Reilly Online Learning. Addison-Wesley Professional. Retrieved September 28, 2023, from https://learning.oreilly.com/videos/amazon-web-services/9780137928521/9780137928521-AWS3_01_01_06/.

[2] Philippe Abdoulaye, "The Digital Economy's Challenges, Opportunities, and Relevance of AWS," in Transforming Your Business with AWS: Getting the Most Out of Using AWS to Modernize and Innovate Your Digital Services , Wiley, 2022, pp.3-16.