**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Final Deliverable - Cloud Computing on AWS**

**Cloud migration project: For XYZ - SA**

**Palash Bhatnagar**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**1. Introduction**

To carry out the migration to the cloud we are going to use a five-step framework to organize the ideas and the steps to follow. We will describe the strategies used in the migration in five main sections throughout this work. Namely:

* 1- Introduction
* 2- Opportunity
* 3- Analysis
* 4- Planning implementation times
* 5 - Migration and results
* 6- Future Optimizations and Continuous Improvement

Additionally, the work has a seventh section of best practices, an eighth section of conclusions and a ninth section of bibliography.

**2- Opportunity**

2.1 - Company description

XYZ - SA is a company dedicated to offering financial services digitally. Its main product line consists of a payment gateway so that different businesses can charge their users without having to develop their own infrastructure. Additionally, users can have their own accounts in the application, saving their funds and transferring them to different accounts of other contacts. The application is accessible via web and mobile. Both access modes query a backend that handles authentication, security, and transactions between accounts and merchants.

Currently, its entire digital infrastructure is set up on premise since at the time the company was founded more than a decade ago, cloud computing services were relatively little used.

2.2 - Context

The company, born in 2006, initially developed its operations in India, but in recent years it began to expand to countries in the region. This expansion, added to the growing popularity of the payment product, has led the company to experience failures in its systems at critical moments, such as when electronic businesses make many requests to the payment gateway's API on dates such as CyberWeek or Black Friday.

The company is looking for a solution to be able to scale the power of its equipment during these critical dates, without having to invest large amounts of capital in infrastructure that will remain idle for most of the year. This is why the company is interested in carrying out a migration process to the cloud.

Also, given the sensitive nature of the data managed by companies, having adequate backups (persistent, secure and redundant) is difficult since it implies the construction of its own data center with different high-cost specialized services (security, special electrical network, etc).

So, it is clear that migration to the cloud offers clear benefits in terms of reduced costs, increased scalability and improvements in security.

**3- Analysis**

3.1 - Solution description

The proposed solution has several AWS services to replicate the architecture that the company has on premise in the cloud. Among them:

* **EC2**

The backend of the main XYZ - SA application is primarily written in Node.js. Currently, the backend runs on the company's own servers, which are not easily scalable and are not protected against adverse events such as power grid failures.

To replicate this backend structure in the cloud, we will create a series of EC2 instances capable of running the modules written in Node.js. Given the generic nature of the backend, we will use the M family of instances offered by EC2. In particular, we will use the m5.xlarge instances, which offer computing power similar to what the company has on premise, with the advantage that they can be easily scaled during demand peaks such as Black Friday.

* **RDS**

To carry out transactions it is necessary to have database systems that support transactions. To do this, the SQL databases that the company currently uses will be migrated to databases, also of the SQL type, but hosted in the cloud through the AWS RDS system.

* **DynamoDB**

User data (identifiers, contact network, etc.) have a nature that favors their storage in noSQL databases. Currently, the company uses databases in MongoDB to store this data. These databases will be moved to DynamoDB, the noSQL database service in the AWS cloud.

* **S3**

Currently, XYZ -.com has several servers dedicated to storing all types of information (its own data lake) within on-premise servers. Managing this data lake without specialized tools becomes tedious and expensive. Additionally, this fact makes the ETL process difficult. This is why it will be decided to migrate the company's data lake to S3, where access can be managed with greater security. Finally, the migration to S3 will also allow you to configure useful tools for building business intelligence reports, such as AWS Athena.

* **YO SOY**

The company has a growing number of employees to provide its services. Managing access to information is becoming exponentially complex with on-premise servers. The AWS IAM tool will be used to solve this problem by creating a series of policies that will help only certain employees have access to certain data.

* **VPC**

To obtain a higher level of security for the different data, a Virtual Private Cloud will be implemented with AWS. This will have different public and private subnets through different availability zones, which will guarantee high availability and privacy of those services that are on the private network without Internet access. Additionally, it will have a load balancer positioned in front of the autoscaling group that will contain the EC2 instances.

* **CloudWatch, CloudTrail & CloudFormation**

So that the c-level executives of XYZ - SA have an overview of the costs associated with the cloud infrastructure, different dashboards will be created through cloudwatch to show the real-time cost and amount of activity (users, users, etc.) that the application experiences.

On the other hand, through CloudTrail, access logs to the AWS API will be generated, which will ensure that the technical leaders of XYZ - SA will be able to have a history of the changes made to the platform by the developers.

Finally, the CloudFormation service will be used to automatically and systematically generate the services that will be used in each of the AWS accounts that will be created for the company (Development, Testing and Production). This strategy is in line with the best practices suggested by AWS, which considers automation a fundamental part of managing processes in the cloud.

3.2 - AWS account parameters

Most of XYZ - SA's services are provided to clients based in Latin America. In general, latency is a determining factor for users of this type of financial services, in which users need to make payments quickly. Given the above, we will choose to configure the AWS account in the Sao Paulo region. While it is true that this implies a higher cost compared to using the Virginia region, it is necessary to highlight that the cost of offering slow service to users is even greater, since it implies dissatisfied customers and a future decrease in the market share of the company.

Furthermore, with the aim of maximizing security, an AWS account will be created for each of the three environments that the company needs, that is, an account for Development, one for Testing and one for Production. This will guarantee an additional layer of security for the company, which in the event of an attack on one of the accounts will have the others unharmed.

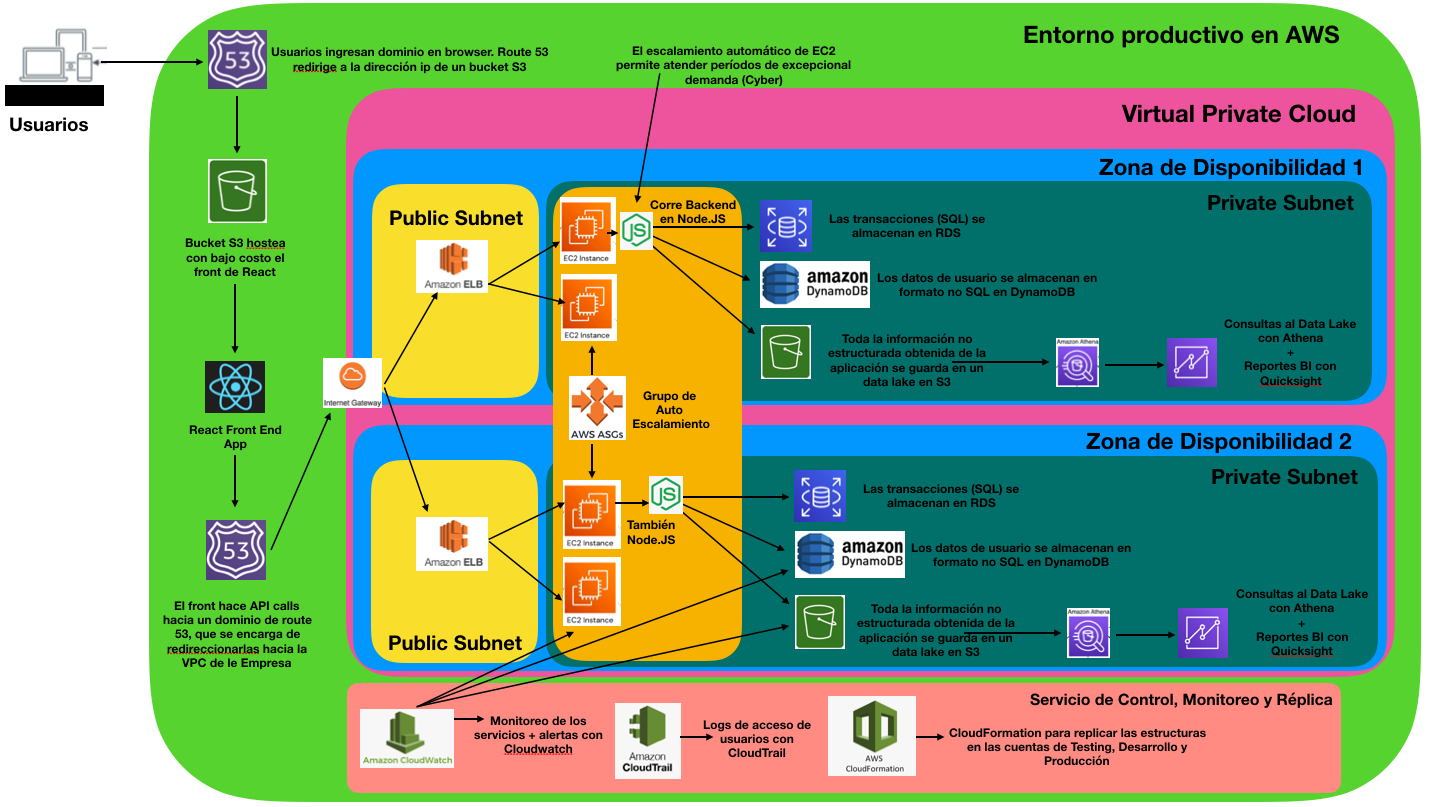
3.3 - Migration Strategies

The main strategy used during the cloud migration process will be REHOSTING, since most of the XYZ - SA applications are packaged and can run without problems on EC2 instances provided with the appropriate environments (Node.js for example). .

3.4 - COE Team

Migrations are complex processes that require a lot of coordination between different people and teams to be successful. The COE team will be in charge of facilitating the migration and will be composed of the infrastructure, security and technical support directors of XYZ - SA

3.5 - Visualization of the solution architecture





3.6 Savings strategies in the cloud

Being a for-profit company, it is of fundamental importance to save costs for the client. For this, the following policies will be implemented:

* The development and testing environments will have less power to save costs. For example, instead of using EC2 m5.2xlarge, the cheaper t3.medium will be used.
* The development and testing environments will be turned off during weekends and between 8 pm and 8 am, since during those hours employees sleep or rest. This represents a saving of approximately 50% in the Development and Testing environments just by better organizing startup and shutdown times.
* The production environment will only be turned on when necessary towards the end of the project, once all other migration processes have been satisfactorily executed.

3.7 - Cost calculator

To estimate the costs of migration to the cloud, the following data provided by XYZ - SA will be taken into account. Currently the company has:

* 30 on-premise servers to run backend services. However, only about 15 are in use. To avoid costs associated with having idle resources, 15 EC2 m5.xlarge instances will be configured. Each will have 10GB of EBS storage to store the resources necessary to run the backend.
* The total storage capacity of the on-premise servers is 250TB, but only about 80TB are being used. Then, a total storage of 100TB will be configured on AWS S3 service. In this way, it will be possible to have enough space for the company's data lake without having high idle capacity.
* The payment platform has 10 million users. Each of them makes about 5 transactions per day, and in each transaction about 100 bytes of information are stored in an SQL database. This implies that about 5GB of new transactions are generated every day. Additionally, for regulatory reasons, transactions from the last three years must be saved so that users can access them. Consequently, 15 powerful RDS m5.large databases will be created, each connected to an EC2 instance running the backend. Each one will have 600GB of storage, which will allow a total of four years' worth of records to be stored in the system (the previous three plus the current one).
* User data that is not a transaction is stored in noSQL format. It is estimated that the information corresponding to each user occupies about 5 KB of storage. Then, a DynamoDB database will be created with 60GB of storage, ensuring there is room for the growth of the application.
* The VPC will be configured with a load balancer capable of handling 700 calls per second, resulting from averaging the 50 million transactions carried out throughout the day, added to a small safety margin to guarantee good response times.
* AWS Athena will be configured so that up to 45,000 queries per week can be made against the company's data lake.
* The development and testing environments, as explained above, will have the same structure (generated through cloudformation templates) but with lower power and startup times, to save costs.

Below is the link to the AWS cost calculator, with all the items described above detailed:

<https://calculator.aws/#/estimate?id=362c0b3caef57771d518ae97996ba46a43924a5b>

As can be seen, the total annual cost of the proposed solution is approximately 282,000 USD per year, which is equivalent to 2 cents per user per month. Although the solution involves a considerable sum of money, when compared to on-premise costs, the advantage of cloud services is clear. Currently, the company has 14 employees, each of them with an average salary of 50,000 USD per year, necessary for the maintenance of the on-premise systems. It is estimated that with a migration to the cloud, only 5 employees will be needed to manage the same number of systems (since AWS takes care of a large part of the workload).

This means that migration to the cloud will require an investment of USD 532,000 annually, compared to USD 700,000 invested annually in fees currently. To this equation we must add the potential benefits of greater security against failures in the electrical system and the advantages of not having a large amount of electronic equipment that depreciates year after year. Finally, moving to the cloud will also allow the company to scale quickly during high-demand dates, such as Black Friday, without incurring large capital investments.

**4- Planning implementation times**

4.1 - Description of the implementation schedule

Obviously, creating all of these cloud services takes a non-trivial amount of time. Both the migration to the different databases for instant consumption (Dynamo and RDS) and the creation of a new data lake in S3 require several weeks of work. Additionally, it is necessary to calculate a certain amount of time for testing once all systems are correctly structured. Here is a six-week schedule for the migration.

* **Week 1:**
  + Creation of AWS account, Root user, IAM users and permissions policies.
  + Create the VPC with public and private subnets in two availability zones.
  + Configuration of access logs through CloudTrail.
  + Creation of EC2 instance and system image to run the backend.
  + Backend functional test on the new EC2 instance.
* **Week 2:**
  + Creation of databases in RDS to model transactions.
  + Creation of databases in DynamoDB to model users' personal data.
  + Backend functionality testing running on EC2 by executing calls to the databases in RDS and DynamoDB.
  + Creation of alerts in CloudWatch and templates in CloudFormation.
* **Week 3:**
  + Creation of different S3 buckets to store permanent type information.
  + Connection testing between the backends running on EC2 and the data stored in S3.
  + General testing of system access and backend.
* **Week 4:**
  + Partial data migration to RDS, DynamoDB and S3.
  + Intensive and automated testing of all backend functions after the final migration.
  + Creation of Athena service to execute queries against the data lake in S3.
* **Week 5:**
  + Total data migration after the previous week's testing.
  + Automated tests in the testing environment.
* **Week 6:**
  + Survey of new systems in production.
  + Review of achieved objectives and setting of new metrics.

4.2 - Graphic Table of the Implementation Schedule

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Month | 1 | | | | 2 | |
| Activity | Week | 1 | 2 | 3 | 4 | 5 | 6 |
| AWS root account creation + IAM users | |  |  |  |  |  |  |
| Creating policies and permissions for developers and administrative employees | |  |  |  |  |  |  |
| Creating EC2 instance with Linux image | |  |  |  |  |  |  |
| Creating VPC with public and private subnets + ELB | |  |  |  |  |  |  |
| Configuring access logs with CloudTrail | |  |  |  |  |  |  |
| Preliminary upload and testing of packaged Node.JS backend on EC2 | |  |  |  |  |  |  |
| Creation of RDS databases to model transactions with SQL | |  |  |  |  |  |  |
| Creation of DynamoDB Databases to model personal data | |  |  |  |  |  |  |
| Backend Connection Testing with the databases | |  |  |  |  |  |  |
| Creating alerts in CloudWatch and Templates in CloudFormation | |  |  |  |  |  |  |
| S3 Bucket Creation | |  |  |  |  |  |  |
| Backend connection testing with S3 Buckets | |  |  |  |  |  |  |
| General testing of system operation | |  |  |  |  |  |  |
| Configuring Athena for BI reports | |  |  |  |  |  |  |
| Data migration from on premise to Dynamo and RDS after successful testing | |  |  |  |  |  |  |
| General testing after migration | |  |  |  |  |  |  |
| Lifting in production | |  |  |  |  |  |  |
| Review of achieved objectives and definition of new goals | |  |  |  |  |  |  |

**5- Migration and results**

5.1 Strategic Results

With the implementation of a system like the one proposed, better results would be achieved in terms of latency during periods of high intensity in the API. Additionally, there would be better data access control thanks to the AWS IAM system. In total, a cost reduction of around 25% is estimated, added to the intangible improvements in scaling capacity.

**6- Future Optimizations**

In the future, it is possible to make different optimizations to the architecture. Between them:

* Leverage various machine learning services offered by AWS, such as Sagemaker, to train models that detect and prevent fraudulent transactions.
* Use tools such as AWS RDS Performance insights to monitor the load on the RDS databases and optimize the design of queries against the database.
* Make use of services such as AWS Elasticaché to cache frequent queries, offering lower latency to users and reducing the impact on RDS and DynamoDB databases.

**7- Recommended Best Practices**

* Carefully save the root user and password of the new system. This user has higher administrator permissions than any IAM user. In this way, a compromise of the root account would represent a great challenge for the integrity and security of the company's computer systems.
* Delegate a team of people within the company to work on, modify and maintain the IAM permissions system. Controlling the security of this type of data is essential, and AWS offers a very powerful system to manage it. It is essential, however, that those who manage this system are trained in its use.
* Tag all created services and put policies in place that force the use of tags with the creation of each service. In this way, in the future it will be possible to filter the different cost centers by labels to make better financial decisions for the company.
* Delete old AMIs.
* Change IAM user passwords every a predetermined amount of time, to strengthen system security.

**8- Conclusions**

Currently, the company faces major obstacles in scaling its IT systems. Migrating systems to the cloud offers in this case an interesting alternative to solve this challenge.

First, the migration will allow servers to be scaled more easily during critical dates such as Cyber ​​or Black Friday. Second, it will lower infrastructure costs by 25% or more, since fewer staff will be needed to maintain the same number of systems. Additionally, AWS is responsible for data center maintenance, redundancy, and hardware security. Third, the migration will resolve data access and security issues through the powerful access and permissions policy system offered by AWS IAM. Fourthly, through S3 it will be possible to build its own data lake from which the company could obtain valuable insights through services such as Athena and Quicksight.

Ultimately, the migration will represent a cost reduction and a competitive advantage for the company, which will be able to scale its infrastructure based on demand. This fact will allow the company to offer better service to its customers during times of high volume, generating more satisfied customers and a probable expansion of market share.

**9- Recommended Bibliography**

* AMAZON (2023). Six advantages of cloud computing. Recovered from:[https://docs.aws.amazon.com/whitepapers/latest/aws-overview/six-advantages-of-cloud-computing.htm](https://docs.aws.amazon.com/whitepapers/latest/aws-overview/six-advantages-of-cloud-computing.html)l
* AMAZON (2023). Cost optimization Pillar - AWS Well-Architected Framework. Recovered from:[https://docs.aws.amazon.com/whitepapers/latest/aws-overview/six-advantages-of-cloud-computing.htm](https://docs.aws.amazon.com/whitepapers/latest/aws-overview/six-advantages-of-cloud-computing.html)l