**Group Project Final Report**

**Key Contributors:**

**Jason Lavenant**

**David Grant**



**4/19/21**

This project and the preparation of this report were funded in part by …. through an agreement with the University of the Incarnate Word. Cyber Security Systems and the University of the Incarnate Word

EXECUTIVE SUMMARY

For our cloud computing project our goal was to develop and host a Local Weather website using Amazon web services. We researched different AWS services and decided which ones would help our website run as efficient as possible. We then created the website and create the Cloud infrastructure. After creation we tested and validated to make sure everything runs smooth.

Project Milestones:

1. Milestone 1 is creating the first design of our cloud and website
2. Milestone 2 is developing the cloud using AWS and deploying it
3. Milestone 3 is finding the cloud resources we can utilize and managing server
4. Milestone 4 is the testing and validation of all our cloud and website

Deliverables:

1. Project status report
2. Development of website and cloud design
3. Deployment of Cloud architecture
4. Operational website hosted on the cloud

Professional Accomplishments: E.g. New skills that you developed

1. The ability to deploy different Amazon Web services cloud resources
2. Create a webserver using Amazon Web Services
3. Maintain a Webserver on Amazon Web Services

# PROJECT SCHEDULE MANAGEMENT

**Gantt chart**

Chart, waterfall chart

Description automatically generated

Project Management Board Link (QR Code Only). Send invite to user: @gonzalodparra

Qr code

Description automatically generated

**Github Project Repository**

https://github.com/Jasonlavaa/Cloud-project

The first step in the design of cloud infrastructure is choosing which AWS services to use. The Auto Scaling Group is best for the project because it allows us to have control on the number of resources we are using.

Graphical user interface, text, application, email

Description automatically generated

The group size option allows to decide our desired, minimum, and maximum capacity of instances we want to have running. The website will always fluctuate in the amount of people that are viewing the website, so it automatically adapts to run as efficient as possible.

Next in the ASG setup we attached a Classic load balancer, this allows the load to be balanced across multiple instances. The Classic load balancer will also run ELB health checks. This allows healthy instances to be deployed and replace instances that fail the health check.

Graphical user interface, text, application, email

Description automatically generated

For the EC2 instances in the Auto scaling Group, for the project the instances we used are Amazon Linux 2 with a type on T2.micro. A security group was configured to allow communication with port 80 (HTTP) and port 22 (SSH). In the user data of the instances, it was defined to install Apache a http service and the html for the website was created in a index.html file.

For the network configurations of the Auto scaling group, we created a Virtual private cloud and attached it the ASG. Along with choosing the different availability’s zones for the instances to run in.

Graphical user interface, text, application

Description automatically generated

A screenshot of a computer screen

Description automatically generated with medium confidenceThis allows the webserver to display the Local weather information.

Graphical user interface, text, application

Description automatically generated

# 

# TABLE OF CONTENTS

Page one: Title Page

Page two: Executive Summary

Page three: Project Schedule

Page four: Project Details

Page five: Project Details

Page six: Project Details

Page seven: Table of contents