**SECURELY DELIVERING DESKTOP APPLICATION TO WEB BROWSER**

JAYASHREE R, GAYETHIRI R.SINBA M, MARRY METILDA S4

Assistant Professor, UG Scholar/CSE-KGiSL Institute of Technology, Coimbatore [saigayethiri@gmail.com](mailto:saigayethiri@gmail.com)

[inba21mani1998@gmail.com](mailto:inba21mani1998@gmail.com)

[reega210@gmail.com](mailto:reega210@gmail.com)

**ABSTRACT**

In this project, we are attempting to securely deliver the desktop application to a web browser and deploy it in a cloud using AppStream 2.0's amazon web services. We have created a desktop application for customer relationship management with information about college management. The aim of this project is to deploy and stream desktop applications using Amazon AppStream 2.0, a fully managed application. Secure application streaming service running AWS Cloud. Provide an Amazon Virtual Private Cloud (Amazon VPC) to deliver an isolated virtual network infrastructure within the AWS Cloud. This environment is going to be used by your AppStream 2.0 resources.

The AWS Management Console used to build an AppStream 2.0 environment to perform the basic administrative tasks required.

1. Use an image builder to install and configure desktop applications for streaming.

2. Provide a fleet of streaming instances for your applications. This fleet will have the standard type of instance and scaling policies to match the number of users who want to be able to simultaneously stream the applications.

3. A stack is creating a web portal from which applications can be streamed by users.

4. Create a user pool to manage users who are able to access your streaming applications.

**Keywords**:Desktop application, Appstream 2.0, AWS Cloud, College Management System

# INTRODUCTION

AWS is a secure platform for cloud service offering,computingpower,database,storagedelivery and other features.

A Desktop Application is a computer program that runs on a computer device locally, such as a desktop or laptop computer, as opposed to a web application that is delivered from a remote server to a local device over the Internet.You can store the desktop applications and share them with the browser. The server will save these applications. It is not possible to provide assurance of secure application use in this system. The data will be shared with the browser more securely without any server problems or other storage problems. The shared applications, data can be secured in VPC services and will be available 24/7. Provide an Amazon Virtual Private Cloud (Amazon VPC) to provide the AWS Cloud with an isolated virtual network infrastructure. This resource for AppStream 2.0 will be using this environment.

The main requirements for starting our project are as follows:

•An AWS account: To use AppStream 2.0 and other AWS services, you first need an AWS account.

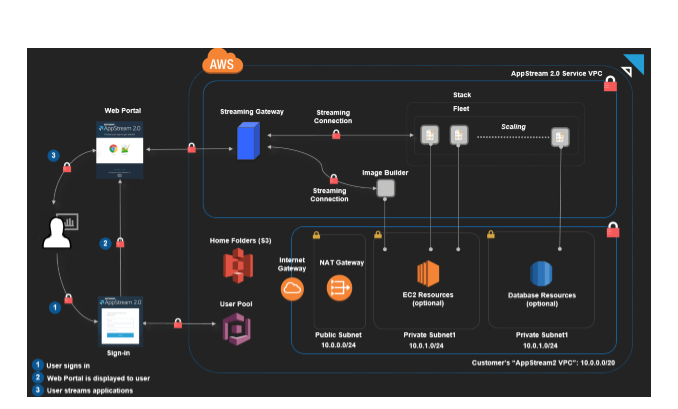
•A current email address: The AppStream 2.0 environment user configuration process is sent by AWS. Users two emails. Complete the process with these emails. Our main goals in this project are as follows:

• To ensure easier and more secure data delivery. • Minimizing the time required. • 24 \* 7 Data access. • Validity

**OUTLINE OF THE PROJECT**

Administrative tasks can be performed in the AppStream console. Creating network resources- create an Amazon Virtual Private Cloud (VPC) and other network resources needed for your AppStream 2.0 environment. Creating an AppStream 2.0 image builder and creating a user pool to manage users accessing streaming applications.

**LITERATURE REVIEW**



AppStream 2.0 is a fully managed, secured application streaming service that enables desktop applications to be streamed from AWS to a web browser HTML 5. Users can use any computer, including Macs, and PCs to access desktop application.Install and configure Apps on AppStream and stream applications to users. Applications are accessed via an HTML 5 browser, and we can control the permissions that can access them. Using the AppStream console it can perform administrative tasks. Use the image builder to install the applications. Optimizing applications with the Image Assistant. It will provide a fleet for desktop applications and provide a stack to stream applications to users. Supply to an Amazon Virtual Private Cloud (VPC) using the AWS Cloud template provided. Within the isolated virtual network infrastructure, this VPC will host the AppStream 2.0 resources.

**SIGNIFICANCE OF THE PROJECT**

• Importing desktop application to create an image builder is important.

• Helps the services to stream.

• Application easy and secure.

**Modules Used In this project**

* Module 1: Network Resources and AppStream Image Builder Creating
* Module 2: Connect to the Image Builder and install the app.
* Module 3: Create AppStream Image.
* Module 4: Fleet provision and Stack and User Management Appstream.
* Module 5: User Experience and Streaming testing.

**PROPOSED SYSTEM**

You can store the desktop applications and share them with the browser. These applications will be saved in the server. This system cannot provide assurance of secure application usage.

**Future Enhancements**

**·** The data will be shared with the browser more securely without any server problems

or other storage problems. · Shared applications, data can be secured and available 24/7 in VPC services.

**Algorithms Used**

* KMS – Key Management System
* AES – Advanced Encryption System

**STEPS FOLLOWED:**

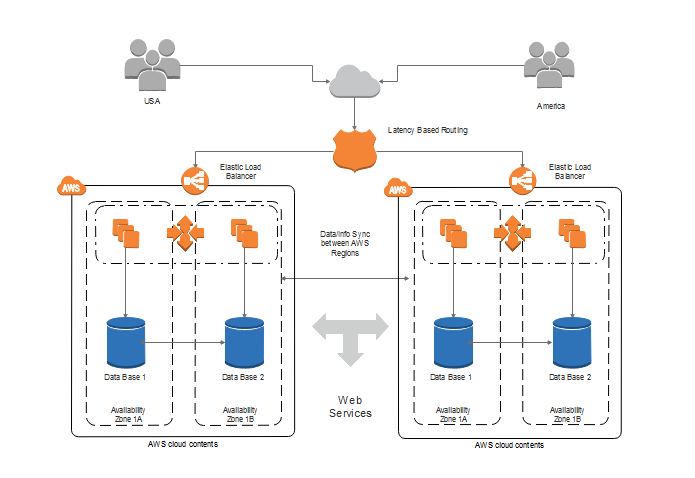
Step 1: Create network resources Step 2: Create an AppStream 2.0 Image Builder Step 3: Connect to Image Builder and Install Apps Step 4: Use Image Assistant to create an AppStream 2.0 Image Builder.

Step 5: Creating an AppStream 2.0 stack and a streaming URL

Step 6: Manage user access with an AppStream 2.0 user pool

Step: Test end - user authentication and streaming experience

Step 8: Take the next step with AppStream 2.0



**CONCLUSION:**

Appendix A: Create and activate an AWS account .

Appendix B: Create and configure network resources manually.

Appendix C: Configure your Appendix 2.0 resources in Chrome.

**Stop and delete your image builder**

1. Open the AppStream2.0 console at

https://console.aws.amazon.com/appstream2

2. Select Images, Image Builder, in

The navigation pane.

3. Confirm if the image builder you

Created. If not, select the builder of

The image and select Actions,

Stop. Repeat this step for every

Image builder you created if you

Created multiple image builders.

4. Choose Actions, Delete after the

Image builder has stopped.

**REFERENCES**

• Amazon AppStream 2.0 Product Details • Amazon AppStream 2.0 Pricing Details

• Amazon AppStream 2.0 FAQs

• Amazon AppStream 2.0 Developer Guide

• Amazon AppStream 2.0 API Reference

• Amazon AppStream 2.0 CLI Reference

• Amazon AppStream 2.0 Try It Now Demo

• Amazon AppStream 2.0 Resources