ReferUp

**Aryan Bhave -** [**abhave@scu.edu**](mailto:abhave@scu.edu)

**Chaitra Boggaram -** [**cboggaram@scu.edu**](mailto:cboggaram@scu.edu)

**Juan Zuluaga -** [**jzuluaga@scu.edu**](mailto:jzuluaga@scu.edu)

**Shaunak Galvankar -** [**sgalvankar@scu.edu**](mailto:sgalvankar@scu.edu)

# 

# ABSTRACT

Colleges and universities have several resources to aid with recruitment, however, they fail to take help from employed students or alumni. ReferUp allows students to connect with them via a web application. These employed students or alumni can assist students with referrals for open positions at their workplace, increasing their chances for an interview and possible recruitment. It is a standardized platform where users can connect with employees from several different companies. In exchange for active users, students/referees get potential placements while referrers could receive company benefits/bonuses. Amazon Web Services and its integrated services helped to create a secure, scalable, and monitored platform that can be deployed with ample feature additions or future improvements.

*Keywords:* AWS, referrals, web application, jobs, recruitment, services

Division of Responsibility/Contribution:

* Aryan - Backend; RDS and Django, Amazon IAM
* Juan - CloudWatch
* Shaunak - Frontend; Django, LinkedIn Authentication
* Chaitra - Amazon SES and Django

# INTRODUCTION

Due to mass layoffs and a depreciating job market, college graduates are finding it increasingly difficult to get hired. This project creates a solution that enables students to connect with alumni or senior students that are available to refer candidates for open positions at their workplace. Current business and employment-oriented online services do not allow students to build a smaller community that can assist with hiring practices and job search. Building such a community would allow available referrers to better understand student portfolios and resumes due to their own experiences in similar programs and classes. Engaging with these similarities will allow a referrer to understand a student's strengths or weaknesses and provide further constructive advice.

ReferUp is designed as a web application that allows students and alumni to create their own profiles, and add contact information to connect with other users. Although the current state does not limit user registration to a specific university, application hosts could restrict environments per university by creating a logical perimeter with the help of student emails and their registered domains. This would allow students from the same university to create communities localized to their own colleges, and hosted separately from others. However, established employment-oriented services do provide their users with a variety of services, and allow them to potentially create a global network. These methods overlook the necessity to create a referrer-student relationship provided by ReferUp.

Hosting the application on a cloud platform, such as AWS, allows us to architecturally scale on-demand and widen implementations from one university to several. The variety of services provided would also aid in scaling features using pre-built AWS APIs, easing the implementation process and assisting in creating a more feature-rich application. Widely established applications would also make it harder for ReferUp to be highlighted as a necessity to students, and such difficulties could be overcome with the creation of a student/new-graduate-specific job board that highlights employed users at workplaces that have listed and open applications. Similar application-employee interconnections exist on platforms like LinkedIn that allow users to have a better experience with applicant-employee communications.

# BACKGROUND AND RELATED WORK

The difficulties experienced while searching for job applications motivated the creation of ReferUp. There is extreme applicant competition that keeps rising every year, and regardless of widespread employment shortages, job seekers will look to gain any advantage to boost their portfolio. The recent mass layoffs at technology companies have flooded the job-seeking market with even more potential applicants that possess working industry level knowledge.

Existing large-scale platforms have successfully helped thousands to find employment with very different working models relative to ReferUp, and similar small-scale but global platforms have small communities that users aren’t able to take complete advantage of. By restricting a small-scale global platform to a localized community such as students and alumni, users will have the opportunity to make full use of available application features and also benefit from community building.

Some small-scale examples include:

* Jobvite: a software and recruitment company that promotes transparency between referring employees and recruiters. Their products allow users to create job invitations on existing platforms, such as Facebook and LinkedIn, and monitor the hiring process per referral.
* Alumni Referral Programs: Several colleges have referral programs where alumni can pass on student information for potential enrollment. These services also provide referrers with rewards or enrolled students with financial-aid.
* University/Campus Job Referral Programs: Colleges allow current employees to refer candidates and earn bonuses. For example, Stanford’s University IT Employee Referrals program rewards referrers with a cash bonus of $2500.

Although these examples do not provide the same service ReferUp does, their similar features and ideologies helped inspire this project. Increasing transparency between referee and referrers and referring students for off/on-campus work creates the core for ReferUp.

# APPROACH

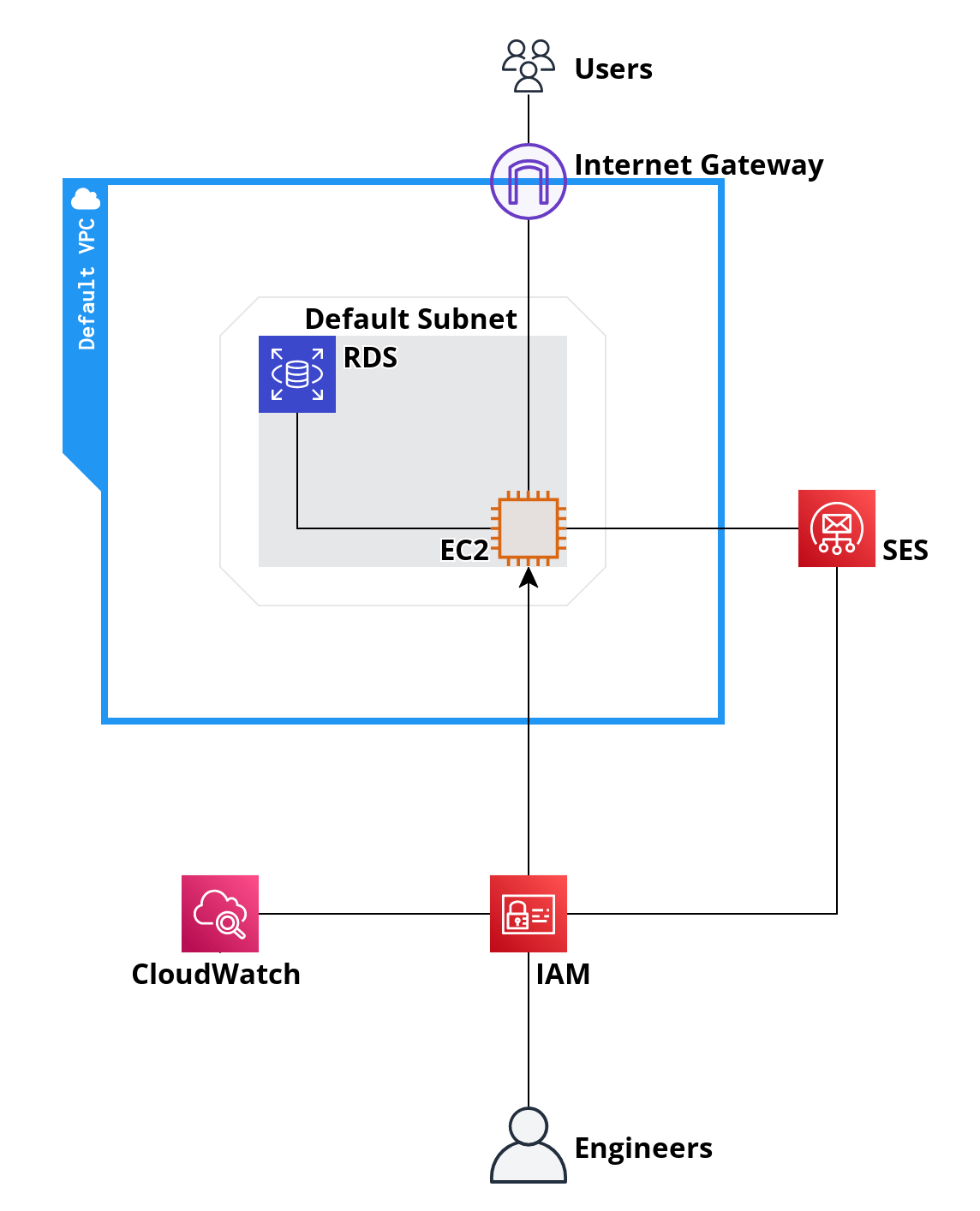
****

Figure 1

The cloud architecture uses the default virtual private cloud (VPC) created by AWS and the default subnets created per availability zone. It contains an EC2 instance and a Relational Database Service (RDS) MySQL database to handle referrer data, authentication data, and session details created with the help of LinkedIn’s authentication API. Figure 1 visualizes the cloud architecture and inbound/outbound links to the VPC.

Generic users can access the EC2 instance that acts as a web server by entering the subnet through an internet gateway located on the logical perimeter of the default VPC. They’re authenticated using LinkedIn’s authentication API and have access to referrer data queried/fetched from RDS. Furthermore, Simple Email Service (SES) enables them to connect with other referrers using a cloud-based Simple Mail Transport Protocol (SMTP) system. Cloudwatch monitors EC2 metrics such as CPU utilization to carry out actions that range from rebooting the instance or using auto-scaling methods to balance usage.

# IAM

Developers and engineers use Identity and Access Management (IAM) to access the cloud architecture. IAM enables easier management of cloud users and increases security by controlling assigned permission policies. A person or program can request IAM to interact with AWS resources. IAM authenticates them against an IAM identity or role, and grants or denies specific permissions. The person or program is then authorized by IAM to perform tasks and actions on cloud resources.

# CloudWatch

CloudWatch acts as an effective monitoring solution and performs actions based on metrics. For ReferUp, it monitors EC2 CPU utilization and notifies developers/administrators if the utilization crosses a predefined threshold.

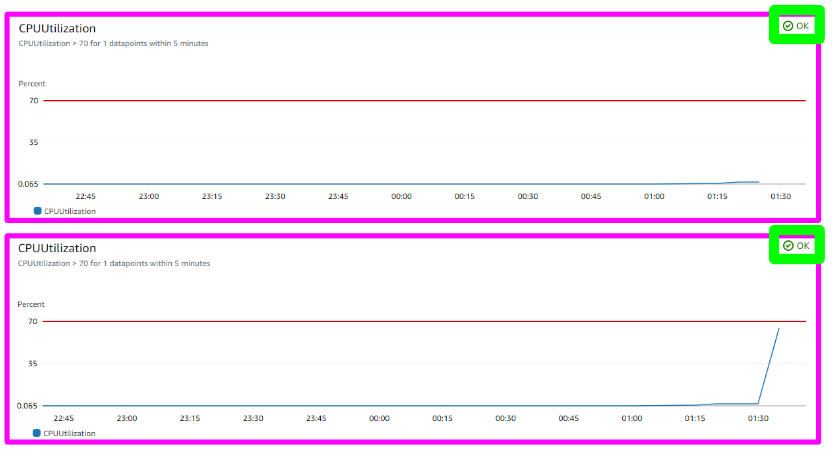


Figure 2

Figure 2 shows CloudWatch monitoring EC2 CPU utilization and reporting an *OK* state with two different metric values. An OK state is reported when CPU utilization does not exceed the 70% utilization threshold.

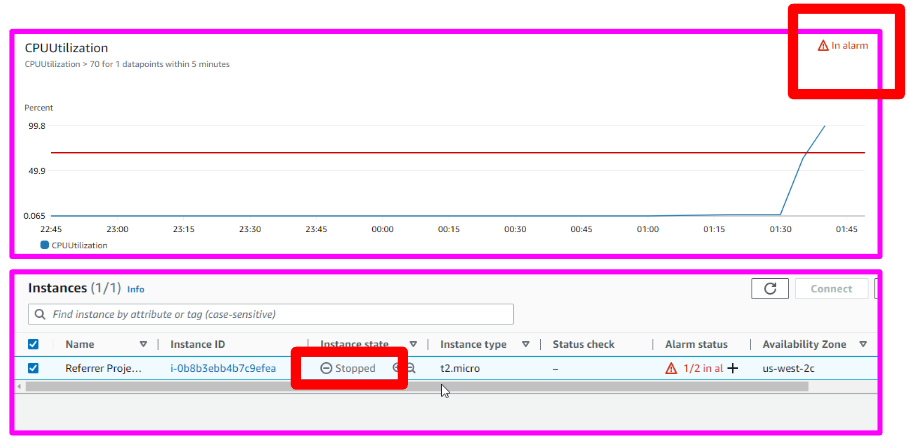


Figure 3

Figure 3 shows CloudWatch monitoring EC2 CPU utilization while running a stress test utility on the EC2 instance, pushing the utilization to cross the 70% threshold and returning an *In Alarm* state. Based on configurations, this forces the EC2 instance to terminate. However, auto-scaling could be enabled to balance utilization/load on-demand.

# SES

SES is connected to an external mail application and creates SES-specific accounts and a sandbox. Accounts that are added to the sandbox can begin to send emails to other SES-verified identities/accounts. Verifying a domain on SES will allow all domain-registered emails to communicate with one another in the sandbox. Emails to unverified identities are prohibited for protection against mass spam. Django allows SES to interconnect with RDS using queries and points the application frontend to email addresses stored in a database table to be used by SES.

# SOFTWARE ARCHITECTURE

Django is used as ReferUp’s primary development platform, and integrates the LinkedIn API with the OAuth2 authorization standard (Auth0).

The project has two apps called “Blog” and “Users”. The Blog app provides a frontend for users to interact with while the Users app handles authentication and session management. Both apps are linked to RDS and either store or fetch data from database tables. Figure 4 shows referrer data stored in RDS. This data is fetched and displayed on the homepage. 

Figure 4

Additionally, OAuth2 has a generic architecture and allows several providers, such as LinkedIn, to create APIs that authenticate users on different websites without the need for passwords.

The Linkedin API has permission levels that give access to user properties such as email addresses, first names, last names, profile pictures, etc. ReferUp makes use of the Lite Profile permissions to fetch data from a user’s LinkedIn account, and can be used by Django to prepopulate form fields, or stored in RDS to be queried.

Forms are used to register users as referrers, and not only allow employed users to be visible as a referrer but also enable other users to change their status to a referrer once they have been hired. Figure 5 briefly highlights this layout.

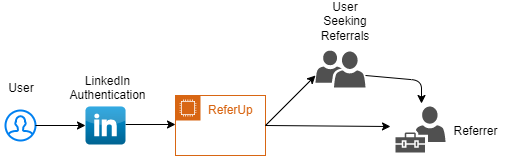


Figure 5

# 

# OUTCOME

Since EC2 instances stop running processes after closing the browser window/tab, a utility called Screen that allows a django server to run in the background without any interference was used. Moreover, specific parameters allow the django server to run on a public IP by creating inbound-outbound rules in the instance’s security group. Using Django’s *makemigrations* and *migrate* options, data stored in a local django managed database was migrated to RDS. This further allows the application to be hosted on different machines without the need for additional configurations and data migration. The EC2 instance type is *t2.micro*, and is connected to an 8GiB Elastic Block Storage (EBS) volume. Observing instance statistics showed a very low level of CPU utilization, however, auto-scaling would be needed with more user interactions.

# ANALYSIS AND FUTURE WORK

The application can be deployed in its current state, however, improvements can better user experience, safety, and engagement. Incorporating more authentication APIs wouldn’t limit users to a specific platform. Currently the application only supports Linkedin, and integrating platforms such as Google, Facebook, Twitter or Instagram would be ideal.

Scaling improvements could also be made to adapt to increasing load/traffic. The current state is built for a single university, and has severe bottlenecks while communicating with a large number of users. Adding more nodes or machines that are connected via a load balancer would prevent any architectural faults or collapses.

To improve user engagement and referral experience, a messaging system would make an enormous difference by creating a one-stop solution and eliminating the need for emails and third-party software. Furthermore, implementing the option to submit resumes to referrers would allow referrers to create a more detailed referral.

# CONCLUSION

Using AWS allows the application to scale vertically and horizontally. Adding features and restricting deployments based on university domains while balancing utilization and traffic with access to a seemingly unlimited pool of resources would assist in creating an industry-functional application. Project limitations were centered around the lack of AWS experience and knowledge of their many software services. However, this did not limit the learning process or the creation of an application using well-documented services. ReferUp enables users to have a seamless user experience while protecting their data as they undertake a tedious job-hunting process and provides a significant advantage to students attempting to enter any highly competitive industry.

# REFERENCES

Auth0. (n.d.). *What is OAuth 2.0 and what does it do for you?* Retrieved December 6, 2022, from https://auth0.com/intro-to-iam/what-is-oauth-2

**PROJECT LINK:** https://github.com/aryanbhave/django\_project