

Background:

A start-up company wants to host its Python and React-based application (Backend: Python API and Frontend React) using AWS. But they are not familiar with the AWS cloud platform. They want to ensure that the application is secure, scalable, highly available, and cost-efficient. As a solutions architect, you have to design a proper solution to meet their below requirements.

Goal:

To architect a solution that is secure, scalable, highly available, and cost-effective using AWS.

Requirements:

- They are concerned about the security of the application, so they have decided to isolate their network from the rest of the customers virtually. Set-up a secure virtual network where the only frontend of application is accessible by users and not the database
- Execute the React application code using AWS Elastic Beanstalk. Ensure that the source code of Web application is automatically picked, pushed to the master branch, and deployed on the servers
- Ensure all the UI images served to the frontend application code are provisioned via a secure storage unit
- There should be enough backups for both the Web and Database server, so if the set-up crashes, we can launch a new one from the disaster recovery backups
- They are uncertain about the traffic pattern that how low or high it can be, so they want the Web application to be running on at least two EC2 instances all time, and when there is a high load, they must burst up to four instances in total
- The Web application should be highly available, even if any VM fails to respond to queries, there should be a mechanism to switch the connection to another healthy VM automatically
- Automate the download all the activity logs into a CSV file, create a stream of data, analyze it, and display it via a dashboard
- The Web application should also be cached globally, so users worldwide can access it with low latency

Deliverables:

- Build the complete solution using CloudFormation, download the .png file and provide the S3 bucket link of images
- Configure all the services and create a document that includes the screenshots of the configured services. Upload the document on the LMS