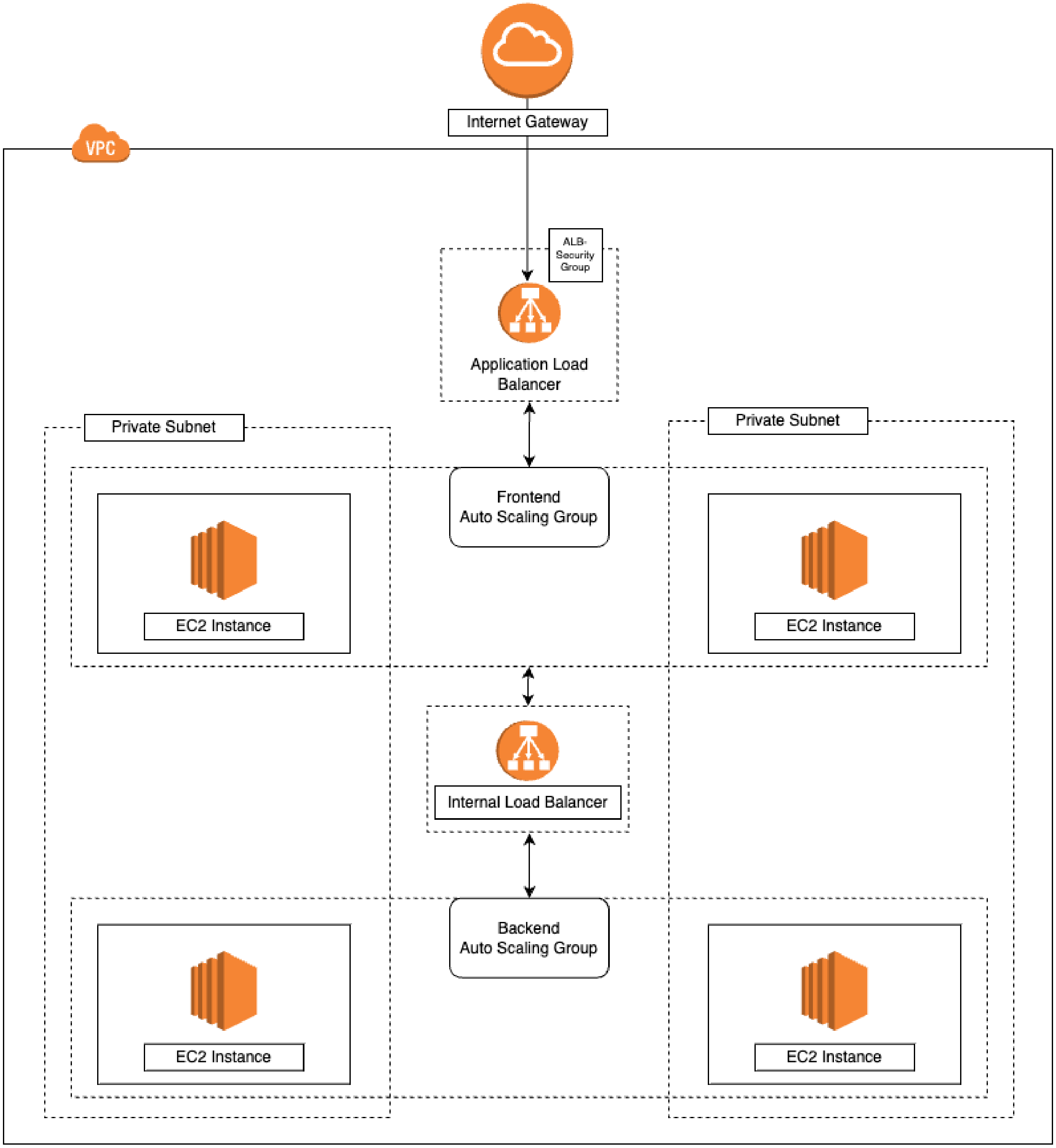
Deployment Design

Executive Summary

This report outlines a comprehensive deployment plan leveraging AWS EC2 instances, Docker containers, and Amazon S3 storage to deploy a robust, scalable application consisting of a frontend Search-UI and an Elasticsearch backend. By employing auto-scaling groups, application load balancers, and carefully managed security groups, the deployment is designed to ensure scalability, high availability, and security. The rationale behind this deployment strategy emphasizes flexibility, control, and the potential for scaling without relying on managed services, despite the inherent challenges of managing such an environment.



Deployment Plan

Infrastructure

- **Compute:** AWS EC2 instances serve as the foundation of our deployment, hosting both the frontend and backend components of our application in

Docker containers. This choice ensures flexibility and direct control over the computing environment.

- **Scalability and Availability:** Auto Scaling Groups are utilized to monitor and adjust the number of EC2 instances dynamically based on demand, ensuring the application remains available and responsive under varying loads. This approach guarantees that our application can handle traffic spikes without manual intervention.
- **Storage:** Amazon S3 is chosen for storing the `cs-valid-dev.csv` file, which is essential for the Elasticsearch service's indexing process. S3 provides a reliable, secure, and scalable object storage solution that integrates seamlessly with our EC2 instances.

Docker Containers

- **Frontend Deployment:** The frontend of our application is deployed using a Docker container image of the Search-UI. This method facilitates easy updates and consistent deployment across all instances.
- **Backend Deployment:** Elasticsearch is also deployed as a Docker container, simplifying the setup and management of the search backend. Post-setup scripts are implemented to automatically index the CSV file from the S3 bucket upon the initial spin-up of each container, ensuring the backend is always ready to serve search queries.

Networking and Security

- **Security Groups:** Security groups are meticulously configured to restrict access appropriately; notably, the Elasticsearch service is not exposed to the public internet and can only communicate with the frontend Search-UI instances. This setup enhances the overall security posture by limiting potential attack surfaces.
- **Load Balancers:** Application Load Balancers are deployed to distribute incoming traffic among EC2 instances, one for the frontend and another for the Elasticsearch backend. This ensures high availability and reliability of the service by preventing any single point of failure.

Rationale

The decision to deploy directly on EC2 instances, as opposed to utilizing managed services like ECS or EKS, offers unparalleled flexibility and control over the deployment environment. This approach, while requiring more initial setup and management, provides the foundation for a highly scalable and available application. It also opens up the possibility for future enhancements, such as adopting a Kubernetes-based deployment model through EKS, which would offer further scalability and deployment efficiencies at the cost of additional complexity and resource overhead.

Alternatives Considered

- **ECS Deployment:** Using ECS for container orchestration could simplify some aspects of deployment and instance management but was ultimately passed over to maintain control over the underlying infrastructure and avoid reliance on managed services.
- **EKS Deployment:** Adopting Kubernetes via EKS presents a forward-looking alternative that could offer superior scalability and flexibility for complex applications. While not chosen for this initial deployment due to its complexity and the need for additional development resources, it remains a viable option for future iterations.

Conclusion

The deployment plan presents a well-considered approach to building a scalable, secure, and highly available application using AWS services, Docker containers, and best practices in network security and application deployment. By emphasizing control over the deployment environment and preparing for future scalability, this plan lays a solid foundation for the application's success and long-term growth.