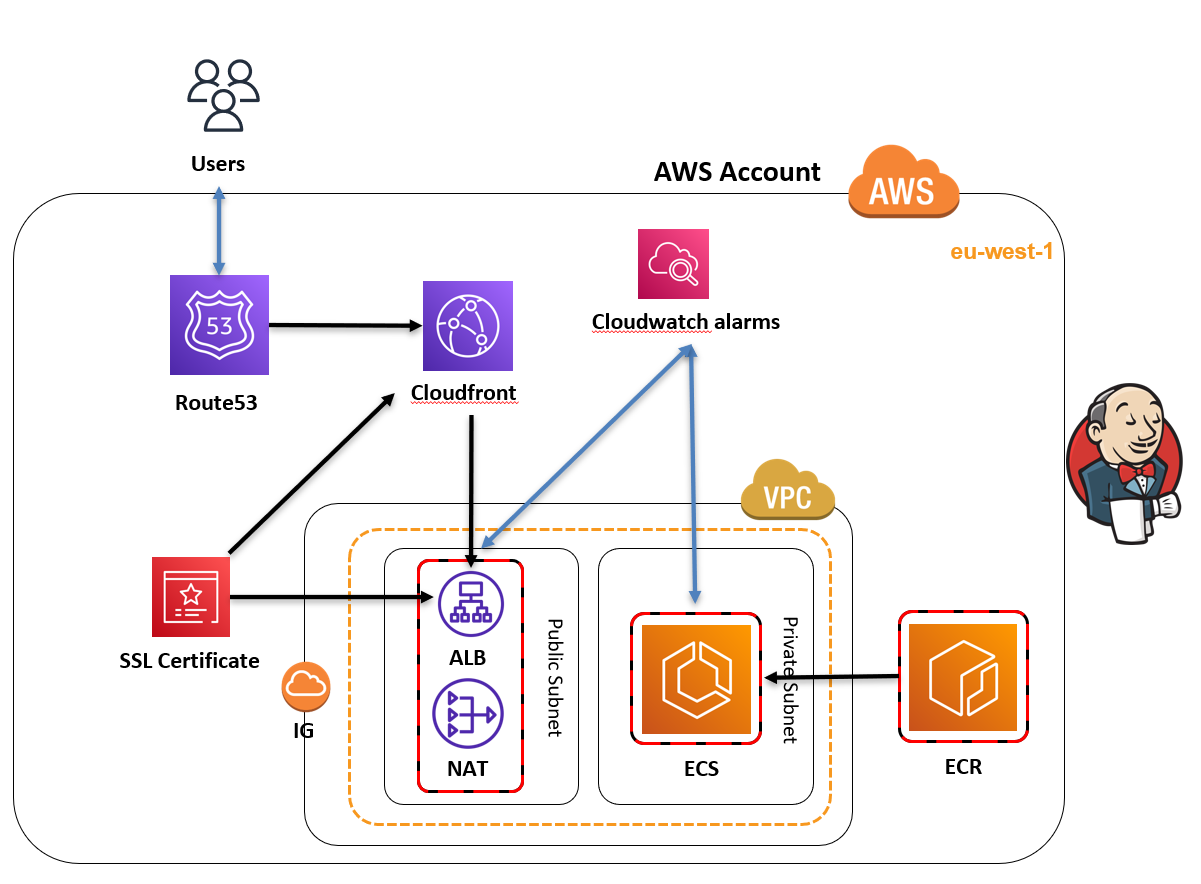
# AWS Architecture for Production Environment

To support the development cycle for the micro-service example provided (<https://github.com/dashersw/microservices-workshop>), the following AWS architecture can be used for the production environment:

1. Amazon Elastic Container Service (ECS): ECS is a highly scalable, high-performance container orchestration service that supports Docker containers. This service can be used to deploy, run and scale the micro-service example in a secure and cost-effective manner.
2. Amazon Elastic Container Registry (ECR): ECR is a fully-managed container registry that makes it easy to store, manage, and deploy Docker container images. It can be used to store the container images of the micro-service example, which can then be pulled and run on ECS.
3. Amazon Elastic Load Balancer (ELB): ELB is a service that automatically distributes incoming web traffic across multiple Amazon ECS instances. This can be used to distribute incoming traffic to the micro-service example to multiple instances, ensuring high availability and scalability.
4. Amazon CloudWatch: CloudWatch is a monitoring service that can be used to monitor the performance of the micro-service example, including CPU and memory usage, request count, and response times.
5. Amazon CloudFront: CloudFront is a content delivery network (CDN) that can be used to distribute the micro-service example to multiple geographic locations. This can help reduce latency and improve the overall performance of the service.
6. AWS Identity and Access Management (IAM): IAM can be used to create and manage users, groups, and permissions for the micro-service example. This can help ensure that only authorized users have access to the service.
7. AWS Certificate Manager: The service can be used to obtain and manage the SSL/TLS certificates required for the service.

Pros:

* The use of ECS and ECR allows for easy and efficient deployment, scaling and management of the service.
* ELB ensures high availability and scalability of the service.
* CloudWatch allows for easy monitoring of the service's performance.
* CloudFront can reduce latency and improve performance for users accessing the service from different locations.
* IAM helps ensure secure access to the service.
* AWS Certificate Manager simplifies the management of SSL/TLS certificates

Cons:

* The NAT Gateway should be kept **deactivated and only activated by Jenkins (or the CI/CD tool)** when needed due to its high hourly cost.
* Less flexibility and control compared to Kubernetes: ECS is a managed service by AWS that offers less flexibility and control compared to Kubernetes which is open source and allows for more customization.
* Less automation possibilities: Kubernetes offers more automation possibilities in managing containers compared to ECS
* Less integration possibilities: EKS being built on top of Kubernetes, it can be integrated with other Kubernetes native tools and solutions while ECS might have some limitations with regards to this.

Challenges:

* Proper monitoring and logging of the micro-service's performance is required to ensure that the micro-service is performing optimally.
* Ensuring compliance with security standards while keeping the solution cost-effective

Potential improvements:

* Use Compute Savings Plans to save up to 40% per month.
* Use AWS Auto Scaling to automatically scale the number of instances based on incoming traffic.
* Use Amazon RDS for database needs instead of self-hosting databases.
* Use Amazon S3 for file storage needs.
* Use AWS WAF to protect the service from web-based attacks.
* Use AWS Shield to protect the service from DDoS attacks.

Note: The budget of $65 per month is quite limited, thus it's important to optimize the usage of the services and resources used in the architecture to stay within the budget. This can be achieved by using the appropriate instance types, optimizing the number of instances used, and using the services in a cost-effective manner. Additionally, it's important to monitor and track the costs of the resources used in order to make adjustments as needed to stay within budget.

