Security Architecture
and
Deployment Architecture
for
ReClothes

Contents

OVERVIEW **2**

SECURITY ARCHITECTURE **3**

Security architecture diagrams 3

Components Description 4

Data Flow in ReClothes 7

INFRASTRUCTURE/DEPLOYMENT ARCHITECTURE **8**

Deployment Architecture Diagrams: 8

Description of Components: 9

Glossary **11**

References **12**

1. OVERVIEW

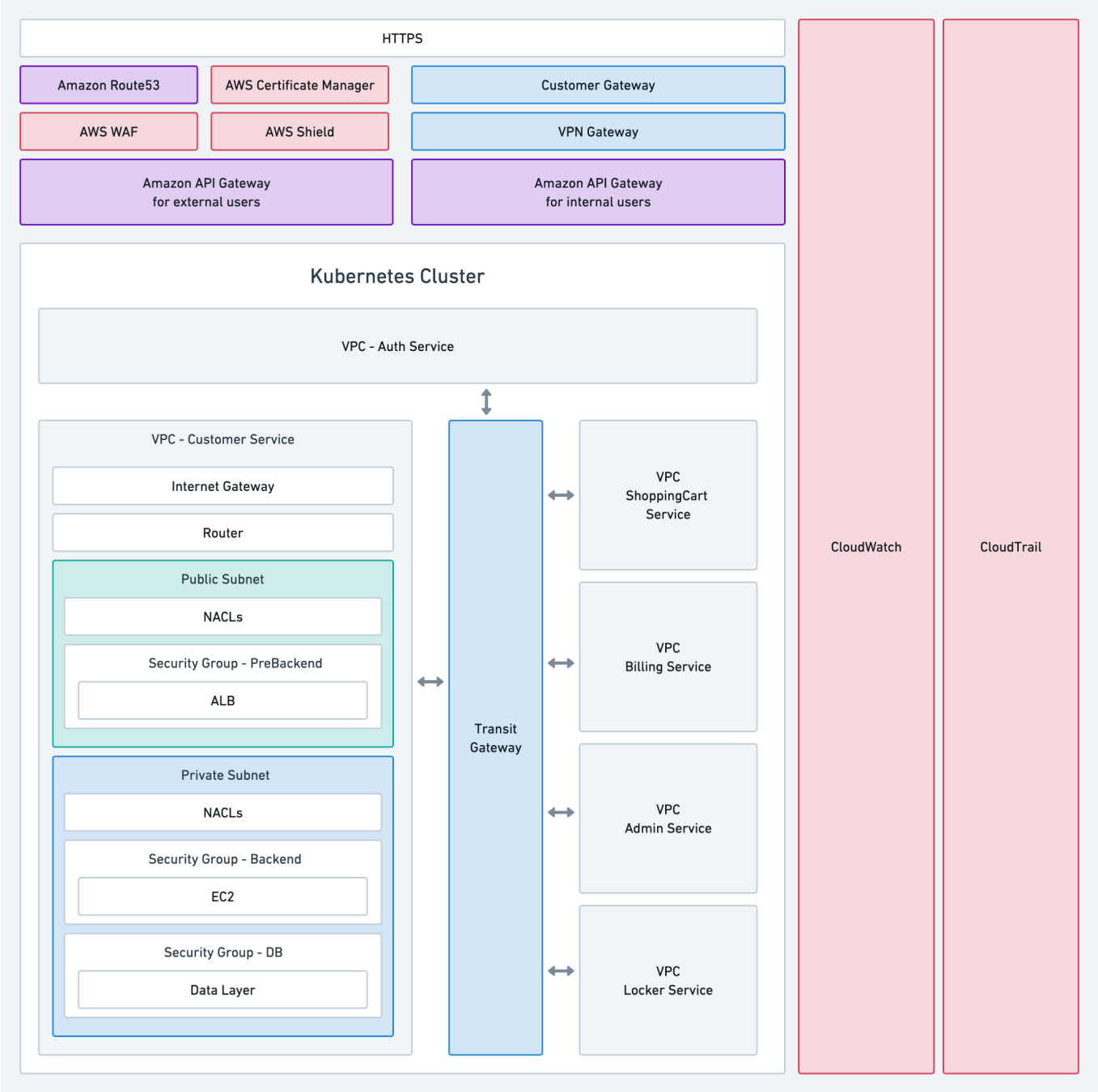
For explaining security among ReClothes, we provide a diagram to illustrate the security architecture. And for further explanation, we also provide a data flow diagram to illustrate requests coming from the outside, and requests sent out to the Internet for OS patching and third-party payment.

In the section on deployment architecture, we use a diagram to illustrate the components we used in AWS and the relationship between them.

Finally, we also provide a glossary table for the explanation of all abbreviations in this document.

2. SECURITY ARCHITECTURE

1.1 Security architecture diagrams



1.2 Components Description

Component	Description
HTTPS	No matter whether requests come from internal users or external users, all requests must be encrypted by HTTPS for security.
Amazon Route 53	Using Route53 as DNS for external users' DNS parsing. Route53 can easily connect with AWS Certificate Manager.
AWS Certificate Manager	AWS Certificate Manager can provide a method for deploying SSL/TLS certificates for HTTPS communication.
AWS WAF	AWS WAF provides features for API Gateway to give the system the ability to detect attack patterns such as SQL injection, Cross-Site Scripting, IP Filtering, and Monitoring.
AWS Shield	AWS Shield is great for DDoS attacks without touching services located in the backend layer.
Customer Gateway	Using Customer Gateway and VPN Gateway can easily establish a connection bridge between the AWS Cloud and the intranet network. By using this, only requests from the company can use the management services. For deliverymen, they need to install a VPN tool on their mobile phones for connecting.
VPN Gateway	It is used with Customer Gateway together for providing a VPN service.
Amazon API Gateway for external users	All external users including buyers and donators will access buying services or donating services with a specific domain name, and this domain name will be parsed by Route53, then generated IP address will point to the external API Gateway. Then API Gateway will validate the authentication of users, if valid, forward this request to a specific service by its request path. For authorization, the API gateway will verify the username and the password by the auth service. If login successfully, the API gateway will send back an access token to the user for later proof. Then subsequent requests will attach this access token for login proof. Each time the API gateway received requests containing an access token from users, it will replace this access token with a JWT for service communication. All services can parse JWT to get some useful information such as session data or time data.
Amazon API Gateway for internal users	Same with external API Gateway. But we will provide a different domain name for internal users including employees, managers, and deliver men. This domain name will be parsed by an internal DNS server, not parsed from Route53.

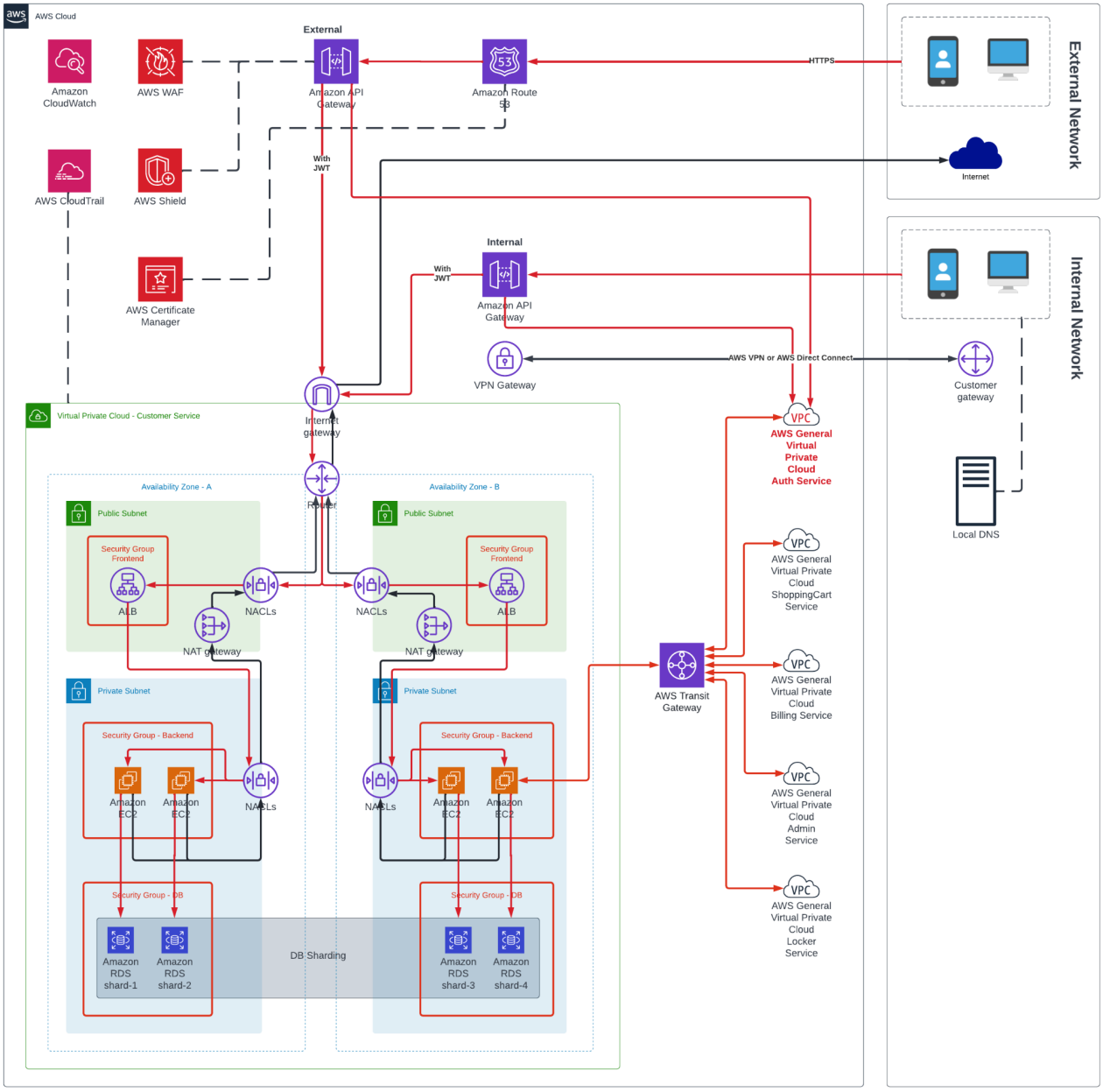
VPC	ReClothes platform uses AWS Virtual Private Cloud to separate services, each service has its own VPC. By using this connections coming in or sending out can be effectively monitored and managed. In addition, each VPC has its own internal IP address range. By using the IP address range, the number of components in VPC can be limited.
VPC - Auth Service	Auth service provides all functions related to authentication and authorization. For Authorization, auth service will compare the data stored in the DB layer with login data received from the API gateway. For Authorization, this only works for internal users since different users with different roles can access different resources.
Internet Gateway	This is the entry of the VPC, all requests coming in or sending out will pass through this gateway.
Router	The router records rules for transferring from one component to another component. By using this, we can easily manage data flow in the VPC.
Public Subnet	All components in the public subnet have permission to connect with the outside.
Private Subnet	All components in the private subnet can't access the outside directly. For some cases, such as OS patching, or third-party payment, the backend services need to access the internet. In this situation, services can connect with NAT Gateway, then NAT Gateway will forward this request to the outside in a limited approach. All sensitive resources will be put in the private subnet, such as computation resources or some database resources.
NACLs	Network Access Control Lists(NACLs) is the core functions of VPC, setting up inbound rules and outbound rules can easily limit known IP address to pass, and other unknown IP address can directly be rejected.
Security Group	Security Group is the supplement of the NACLs, by using it the components located in it can only allow legal ports to access. Furtherly, there's are three kinds of Security Groups in the VPC. One for the components located in the public subnet, another for the compute components located in the private subnet and the last one for the database resources in the private subnet.
ALB	All coming requests must flow into ALB firstly, then forward requests to targets equally. And ALB provides two kinds of routing algorithms, one is the round-robin algorithm, and another is the Last Outstanding Requests algorithm. By using the round-robin, the requests will be forwarded to targets alternatively. And by using the LOR algorithm, requests will be always sent to the target with lower pressure. In addition, ALB has the ability to scale out and scale in automatically based on the traffic of the network.

EC2	Elastic Cloud Compute(EC2) is our core computation unit, and the Kubernetes cluster is also created based on EC2. For higher security, all EC2 instances are put in the private subnet.
Data Layer	Same with EC2, all databases are put in the private subnet for higher security. And through the security group, there's only one way to connect them from the security group in the EC2 instance. AWS also provides HTTPS connections for database services such as RDS, DocumentDB, or ElasticCache.
Transit Gateway	As all services in ReClothes are separated into different VPCs, instances can't connect with each other directly. Transit Gateway can effectively transfer messages from one server in a VPC to another server in a different VPC in a secure way.
CloudWatch	CloudWatch provides all kinds of metrics of AWS components, and we can set some alarms based on a specific value of a metric.
CloudTrail	CloudTrail provides all logs of operations in the AWS account.

1.3 Data Flow in ReClothes

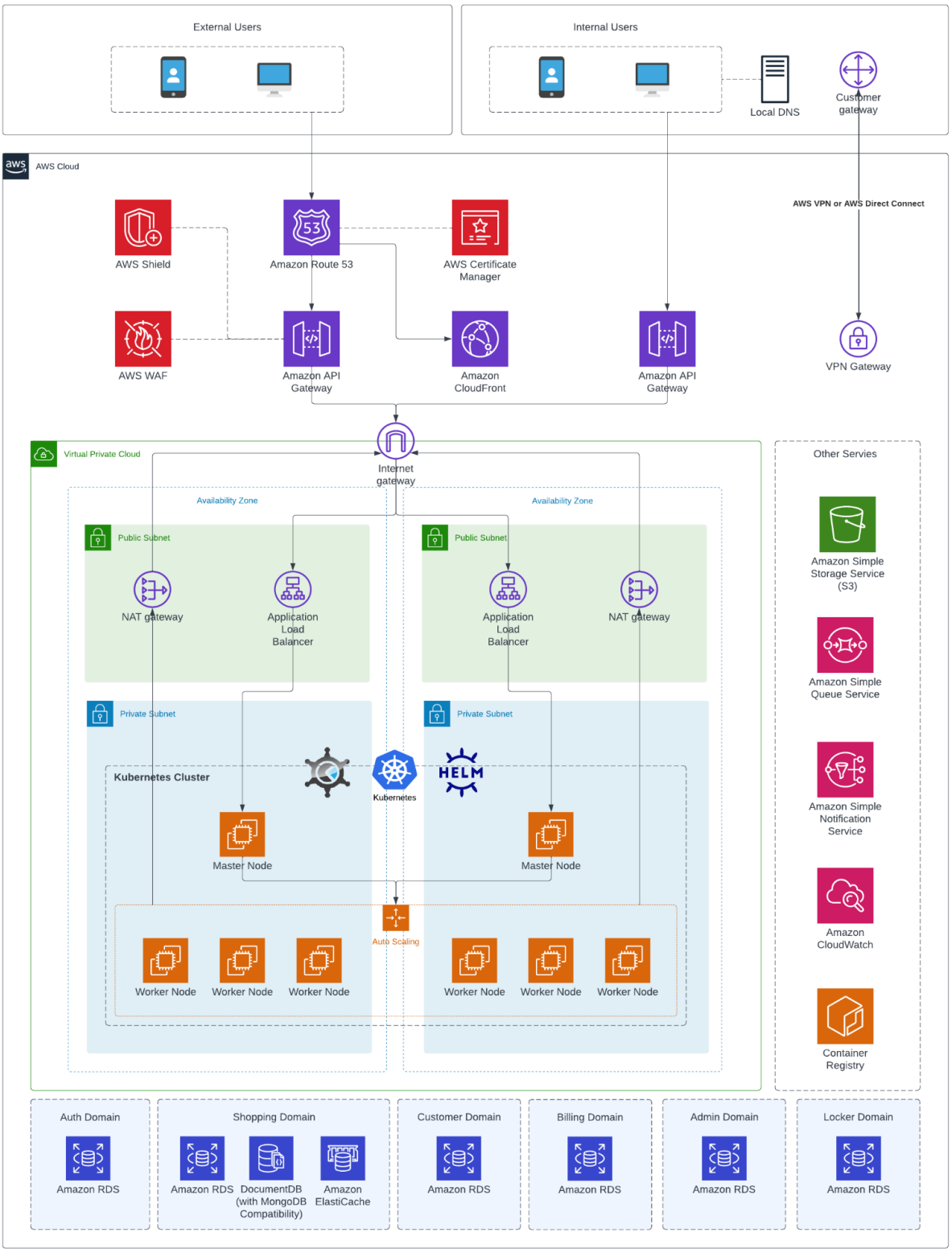
Red Line: represents the requests from the outside

Black Line: represents the requests to the outside (eg. OS patching, third-party payment service)



3. INFRASTRUCTURE/DEPLOYMENT ARCHITECTURE

3.1 Deployment Architecture Diagrams:



3.2 Description of Components:

Component	Description
Kubernetes Cluster	By using the Kubernetes cluster, deploying services can be more fine-grained. One virtual machine can deploy one or more services in isolated namespaces. It's way more cost-efficient than deploying one service in one instance. In addition, Kubernetes can easily grab a Docker image from a Docker image registry to build a pod. And we can set some rules for automatic recovery from a failed condition. Finally, the best part of Kubernetes is that you can write your Kubernetes scripts once, then deploy the Kubernetes cluster in any cloud provider since almost all cloud providers support Kubernetes.
KOps	KOps provides the easiest way to create and manage a Kubernetes cluster in the AWS environment. KOps can help you to map the components in Kubernetes with the components in AWS. For example, the ingress node in the Kubernetes represents the Elastic Load Balancer (ELB) in the AWS. The Horizontal Pod Autoscaler in Kubernetes represents the Auto Scaling Group in the AWS. By using this, we can either use the Kubernetes command-line to monitor or manage the cluster or use the AWS console to manage visually.
Helm	Without using Helm, the installation and destruction is not easy thing in the Kubernetes cluster, you need to apply your script files one by one. After using Helm, you can distinguish your services by alias named in Helm, and use this alias to create or destroy your services.
RDS	RDS is a managed SQL database service provided by AWS. It is great for some situations with strong consistency requirements. In the console, we can use tools to monitor and manage relational databases easily. Domains that used RDS: Auth Domain, ShoppingCart Domain, Customer Domain, Billing Domain, Admin Domain, Locker Domain
DocumentDB	For our shopping cart domain, all products information is stored in this document database as it can give us a great feature of schemaless. By using it, all information related to the product can store in this database, and don't need to worry about different data schemas for different categories of products. Domains that used DocumentDB: ShoppingCart Domain
ElasticCache	AWS also provides a managed in-memory database. We used this for our shopping cart domain and basic configuration information. Since our shopping cart module should provide high availability for customers, it should respond to customers at any time. The in-memory database is a great choice for this kind of requirement. So, all operations related to shopping cart (adding, changing, or deleting

	products in shopping) will happen in this in-memory database. And for fault tolerance and persisting our shopping cart data, we also use a message queue to help us store data into a relational database asynchronously. Domains that used ElasticCache: ShoppingCart Domain, System Configuration
Simple Queue Service	SQS is a managed message queue service in AWS. Using it can separate the system into different modules with lower coupling and it helps to establish a system with asynchronous messages between modules. In Reclothes, some operations may spend more time to finish, in this situation, we can grab some trivial tasks from it to process them asynchronously. And end-users will not be bothered by waiting too long.
Simple Notification Service	SNS can either notify applications or notify people. For notifying people, we can notify them through email, SMS, or mobile push notification. By notifying the application, it can connect with some other services in AWS easily. By using it, we can easily establish a complex function automatically. In ReClothes, we use it for transferring messages asynchronously. For example, for some time-consuming tasks in order creation, we use SNS to process them later. So, end-users will not be bothered by waiting too long. In addition, it can also help us to establish a system with lower coupling.
Container Registry	Each time developers finished features, they can submit codes into a Git repository. Then the Jenkins server can grab codes from it automatically and do some unit tests or integration tests, after that, Jenkins will create a new Docker image for the Docker Image Registry. AWS provides Container Registry for storing Docker images. As long as the Kubernetes cluster creates a new pod or recovers a service, it will grab the Docker image for this Container Registry.
Simple Storage Service	S3 is a great choice for storing static files. Since we have a website for buyers and donators and also have a website for internal management. Some static files, such as images, CSS files, and JS files, we can store in S3 and access in a REST way. For improving the performance of accessing static resources, we can use CloudFront to grab them with lower latency.
CloudFront	By using it, users can access static resources from the edge point close to them, and don’t need to get from S3.

4 GLOSSARY

Term	Description
AWS	Amazon Web Service
S3	Simple Storage Service
RDS	Relational Database Service
K8s	Kubernetes
EC2	Elastic Cloud Compute
ELB	Elastic Load Balancer
ALB	Application Load Balancer
NLB	Network Load Balancer
VPC	Virtual Private Cloud
NACLs	Network Access Control Lists
WAF	Web Application Firewall

5 References

1. James Beswick | 18 Sep, 2019 | Architecting multiple microservices behind a single domain with Amazon API Gateway | https://aws.amazon.com/blogs/compute/architecting-multiple-microservices-behind-a-single-domain-with-amazon-api-gateway/
2. Puru | 20 Jul, 2020 | Design Secure & Scalable AWS VPC for Micro-service Architecture in AWS cloud | https://dev.to/ptuladhar3/design-secure-scalable-vpc-for-micro-service-architecture-4iah
3. Vivek Raju | 23 May, 2019 | Migrating Applications from Monolithic to Microservice on AWS | https://aws.amazon.com/blogs/apn/migrating-applications-from-monolithic-to-microservice-on-aws/
4. Connect your VPC to other networks | https://docs.aws.amazon.com/vpc/latest/userguide/extend-intro.html
5. VPC Peering | https://www.kentik.com/kentipedia/what-is-vpc-peering/
6. Bikram | 3 July, 2021 | VPC Peering explained | https://bikramat.medium.com/vpc-peering-explained-c2d50d85935b
7. Eric Johnson | 9, July 2021 | Integrating Amazon API Gateway private endpoints with on-premises networks | https://aws.amazon.com/blogs/compute/integrating-amazon-api-gateway-private-endpoints-with-on-premises-networks/