

Technology Stack, Architecture Execution plan and Operationalization for ReClothes Terrier

Contents

OVERVIEW	2
Technology Stack	3
Architecture Execution plan	8
Operationalization (Support, Monitoring, Maintenance, etc)	9
4.1 System Monitoring & Log Management	9
4.2 Kubernetes Cluster Monitoring	10
4.3 CI/CD Flow	11
4.3 Agile Management	12

1. OVERVIEW

This document describes the technology choices made in context of the solution architecture developed for the SITAC competition problem statement. The following section provides details about the technologies selected for various layers as well as the reasoning for the selection. In the next section after that, we have provided a high-level plan for implementing the solution. This takes into consideration the dependencies, resource constraints and various other factors.

2. TECHNOLOGY STACK



Architecture component	Technology Choice	Justification
Presentation Layer	Vue.js	Easy to use and the learning curve is not too much steeper.
	Bootstrap	For the internal management portal, Bootstrap provides rich free templates. In addition, Vue.js works better with Bootstrap, using them together can save us a lot of time.
	IOS	Using Swift or Objective-C to develop an IOS app.
	Android	Using Java to develop an android app.
Service Layer	Spring Cloud Framework	Spring Cloud Framework exists for a long time and is more mature than other microservices frameworks. In addition, it's the best choice in the Java environment. Tools used in Spring Cloud: <ul style="list-style-type: none"> • Eureka (Discovery Service) • Spring Cloud Config (Config Service) • Spring Cloud Gateway (Gateway Service) • Spring Cloud Sleuth (Sleuth Service) • Resilience4J (Circuit Breaker) • OpenFeign (Service Communication)
	Spring Boot Framework	Less configuration and works better with Spring Cloud Framework.
Infrastructure Layer	Kubernetes	Great for containers orchestration. We can deploy different services with limited physical resources. Using pods as an extra abstraction layer to isolate different services.
	KOps	Using KOps to deploy the Kubernetes cluster is much easier in AWS, and all Kubernetes services convert to AWS managed services. For example, Kubernetes ingress can convert to AWS load balancer, the Kubernetes Horizontal Pods Autoscaler can convert to Auto Scaling Group in AWS, and we can manage Kubernetes nodes in the AWS EC2 console directly. In addition, CloudWatch can help us to collect all metrics in EC2, ELB, or ASG.
	Helm	By using Helm, we can combine related Kubernetes scripts into one namespace and install or uninstall with one command.

	Terraform	Through Terraform, all operations of setting up environments can store as codes. It's great for maintenance and version control.
	EC2	The Kubernetes cluster uses EC2 to provide resources for the cluster.
	ASG	The Horizontal Pods Autoscaler works as an Auto Scaling Group in AWS. We can use CloudWatch to collect related metrics to monitor the performance of the cluster.
	ALB	The Kubernetes ingress components work as Elastic Load Balancer in AWS. Because the ReClothes platform provides services based on the website and the app, Application Load Balancer(ALB) is the best choice for us. In addition, we can combine this service with WAF to improve prevent common attacks from the outside.
	API Gateway	AWS API Gateway stands ahead of our backend services, it can avoid sudden heavy loads from the outside. It provides some useful features like traffic throttling, working better with OAuth service, request logging, and common security features (XSS, CSRF).
Message Queue	SQS	AWS managed message queue services. For using this, we don't need to worry about the nodes of the message queue cause it's serverless. For processing some tasks asynchronously, SQS can send messages to a specific service without blocking users.
	SNS	SNS can help backend services to send messages to emails, mobile text, and mobile push notifications. It can work with SQS to fan-out messages.
Database Layer	Aurora PostgreSQL	Compared with MySQL, PostgreSQL is better at processing complex, high-volume operations. Using the Aurora PostgreSQL cluster can reduce the time and efforts of management. In addition, Aurora is a cost-efficient choice compared with RDS.
	DocumentDB	AWS DocumentDB is a NoSQL database with the feature of schemaless. So, it's great for the product data because of the different kinds of clothes we collected from donators.
	Elasticache	For acquiring high availability in the shopping cart, Elasticache is the best choice

		as it queries data from the memory, not from the disk. We can use the hash data type to store all shopping cart data.
	S3	For storing static resources, S3 provides the feature of storing and related REST APIs.
	CloudFront	CloudFront is the CDN service in AWS. It works better with S3. After using it, users can get a better experience on our website or apps.
Monitoring Layer	CloudWatch	In AWS, CloudWatch can collect all kinds of metrics from different AWS services. For example, we can collect CPU usage information from EC2, request counts from ALB, or total connection counts from RDS. In addition, it can create some alarms to notify people after reaching a specific value.
	CloudTrail	CloudTrail can provide the feature of auditing. In its dashboard, we can monitor all actions executed by a specific person at a fixed time.
	ELK Stack	Collecting business logs from microservices, it's hard because log files are located in different nodes. By using ELK(Elasticsearch + Logstash + Kibana), we can collect and visualize all business logs in real-time.
	FileBeats & PacketBeats	FileBeats can help us forwards all log files in different nodes to a specific Elasticsearch service. PacketBeats can help us to collect all metrics by a specific protocol. For example, we can collect metrics by a specific port in the HTTP protocol.
	Zipkin	For microservices, it's hard to debug across all kinds of services. By using Spring Cloud Sleuth, it can chain all related services by a request call. For visualization, Zipkin can display a request chain by showing all related services in sequence.
	Prometheus	Prometheus is a kind of time-series database and it is better to collect metrics in a Kubernetes cluster.
	Grafana	After collecting metrics in a Kubernetes cluster, Grafana can show a custom dashboard to display the status of the cluster and all configured metrics.
DevOps Layer	Jenkins	Jenkins is the best choice for CI/CD. It can create a CI/CD pipeline to improve the

		speed of development and deployment. After committing your codes into Git, the hook of Jenkins can be triggered to execute unit testing, then do checking from Sonar Lint, and generating a Docker image, finally deploy it into the Kubernetes cluster. It's totally automatic after creating a CI/CD pipeline.
	Docker	Using Docker to create a Docker image, we send our Docker image to Elastic Container Registry (ECR) for storing.
	ECR	AWS ECR is the best choice for storing Docker image in AWS.
	GitLab	GitLab is the open-source version of GitHub. It also provides the pull-request feature and developer management.
	Git	Git is great for code version management.
	Nexus	Nexus can help us to create a private Maven Repository in an internal network without worrying about security issues.
	SonarLint	After committing codes, codes must filter through SonarLint to check code style, common mistakes, or vulnerabilities.
	JUnit5	JUnit 5 provides annotation and some other useful features to simplify the process of unit testing.
	Mockito Mocking Framework	For API testing, some APIs might rely on other APIs, Mockito can provide a mocking method to generate fixed feedback.

3. ARCHITECTURE EXECUTION PLAN

For launching ReClothes platform to the market as soon as possible, we split the whole platform into two phase1. In the phase 1, we focus on developing the all core backend services, partial internal management features, and the IOS app. It lasts for about 100 days (about a half year). In the phase 2, we focus on optimizing the performance of all backend services and developing the android app. It lasts for about 65 days (about 3 months).

In each project of these sub-projects(Project A-E), we use Scrum to manage each development team. In the beginning, product owners and senior developers need to discuss product backlogs and grab some features from it to generate one sprint for the next two weeks. In each team, all members should report their progress daily and do some code reviews per week. Most importantly, at the end of the sprint, all team members should gather together and do a retrospective meeting to improve performance in the next sprint.

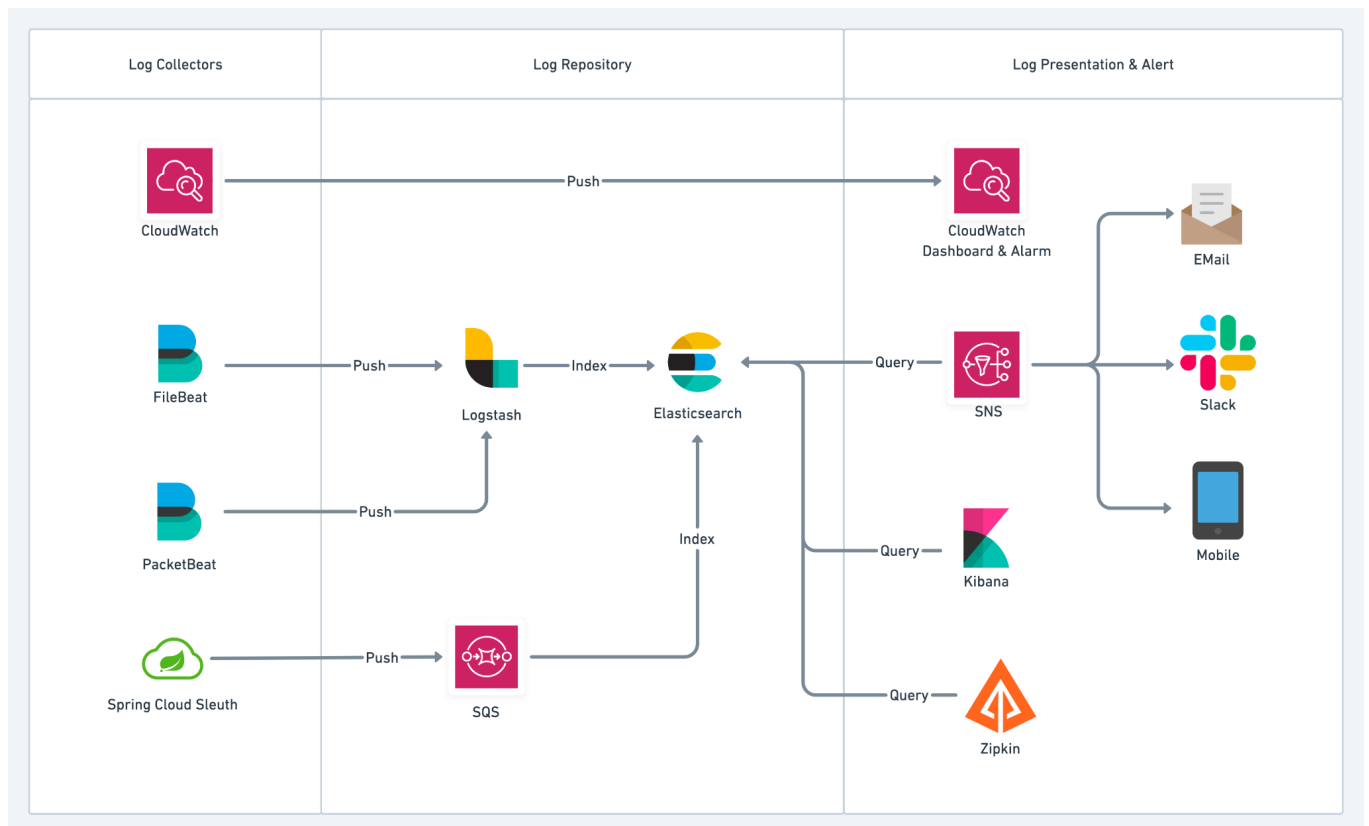
Name	Start Date	End Date	Duration	2022				2023
				Q2	Q3	Q4	Q1	
Kick Off	May 02, 2022	May 03, 2022	2 days					
Stakeholder Interviews	May 02, 2022	May 20, 2022	15 days					
User Stories	May 09, 2022	May 20, 2022	10 days					
▼ Phase 1	May 09, 2022	Sep 23, 2022	100 days					
User Stories Analysis	May 16, 2022	May 18, 2022	3 days					
▼ UI Design	May 19, 2022	Jun 10, 2022	17 days					
Mobile IOS UI Design	May 19, 2022	Jun 10, 2022	17 days					
Internal Management Portal UI Design	May 19, 2022	May 27, 2022	7 days					
Environment Initialization	May 09, 2022	May 20, 2022	10 days					
Project A: Backend Services	May 23, 2022	Aug 26, 2022	70 days					
Project B: Mobile IOS Development	Jun 27, 2022	Aug 19, 2022	40 days					
Project C: Frontend Internal Management Portal	Jul 18, 2022	Aug 12, 2022	20 days					
Backend Services Stress Test	Aug 15, 2022	Sep 16, 2022	25 days					
Integration & Stakeholder Satisfaction Test	Sep 12, 2022	Sep 16, 2022	5 days					
Launching	Sep 19, 2022	Sep 23, 2022	5 days					
Training	Sep 22, 2022	Sep 23, 2022	2 days					
▼ Phase 2	Oct 24, 2022	Jan 20, 2023	65 days					
User Stories Analysis	Oct 24, 2022	Oct 28, 2022	5 days					
▼ UI Design	Oct 31, 2022	Nov 18, 2022	15 days					
Mobile Android UI Design	Oct 31, 2022	Nov 18, 2022	15 days					
Environment Initialization	Oct 24, 2022	Oct 28, 2022	5 days					
Project D: Mobile Android Development	Nov 14, 2022	Jan 06, 2023	40 days					
Project E: Backend Service Optimization	Nov 28, 2022	Dec 23, 2022	20 days					
Backend Services Stress Test	Jan 02, 2023	Jan 06, 2023	5 days					
Integration & Stakeholder Satisfaction Test	Jan 09, 2023	Jan 13, 2023	5 days					
Launching	Jan 16, 2023	Jan 20, 2023	5 days					
Training	Jan 18, 2023	Jan 20, 2023	3 days					

4. OPERATIONALIZATION (SUPPORT, MONITORING, MAINTENANCE, ETC)

4.1 System Monitoring & Log Management

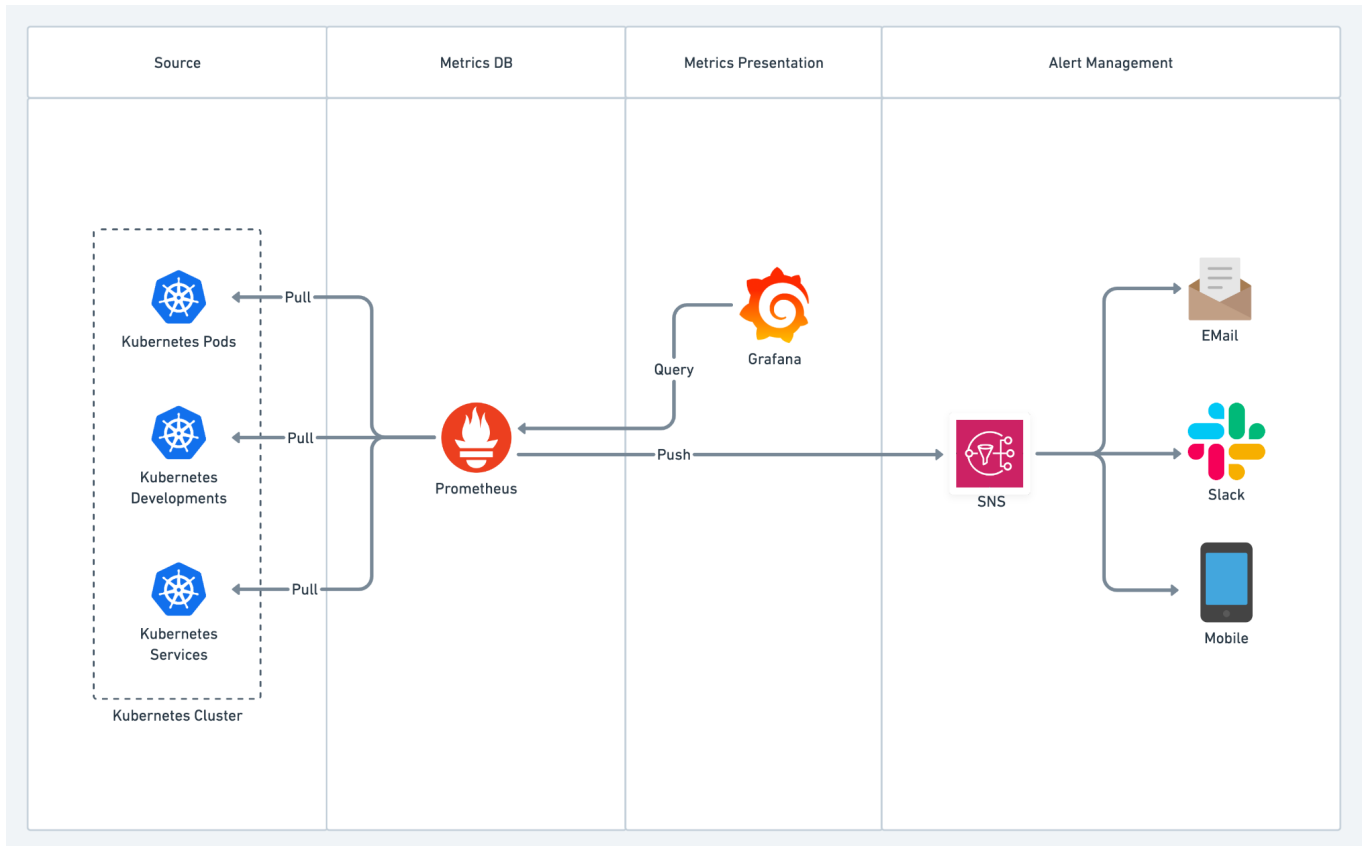
There are three kinds of logging requirements in ReClothes platform.

1. AWS services logging.
For collecting all metrics of AWS services, we can use CloudWatch to monitor and even create alarms for notifications after reaching a specific value.
2. Business logs.
For resolving the difficulties of collecting all business logs from all services, we need to set up a central logging service. ELS stack is the best choice for this. Logstash can collect all log files created by FileBeats or PacketBeats, then data can forward to Elasticsearch for indexing. Finally, developers can check logs from Kibana. In addition, SNS can connect with Elasticsearch for sending notifications to the email, the Slack, or the mobile phone.
3. Request chain.
Spring Cloud Sleuth can generate a unique tracking id for chaining all related services, then Zipkin can help us to visualize this whole request chain for debugging purposes.



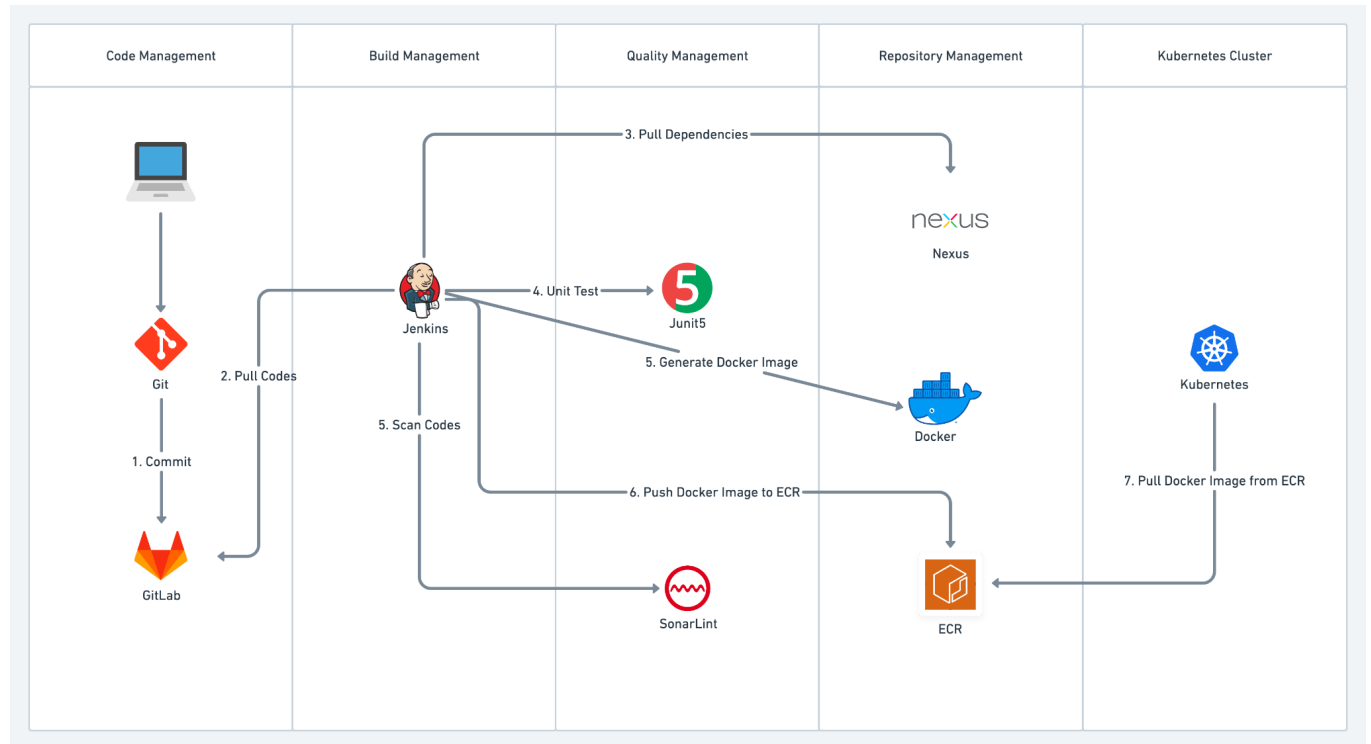
4.2 Kubernetes Cluster Monitoring

Prometheus + Grafana can provide real-time monitoring of the Kubernetes cluster.



4.3 CI/CD Flow

Using Jenkins to create a CI/CD pipeline.



4.3 Agile Management

Each sprint lasts about 2 weeks.

